

Monitoring

Date published: 2024-01-01

Date modified: 2024-12-05



Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

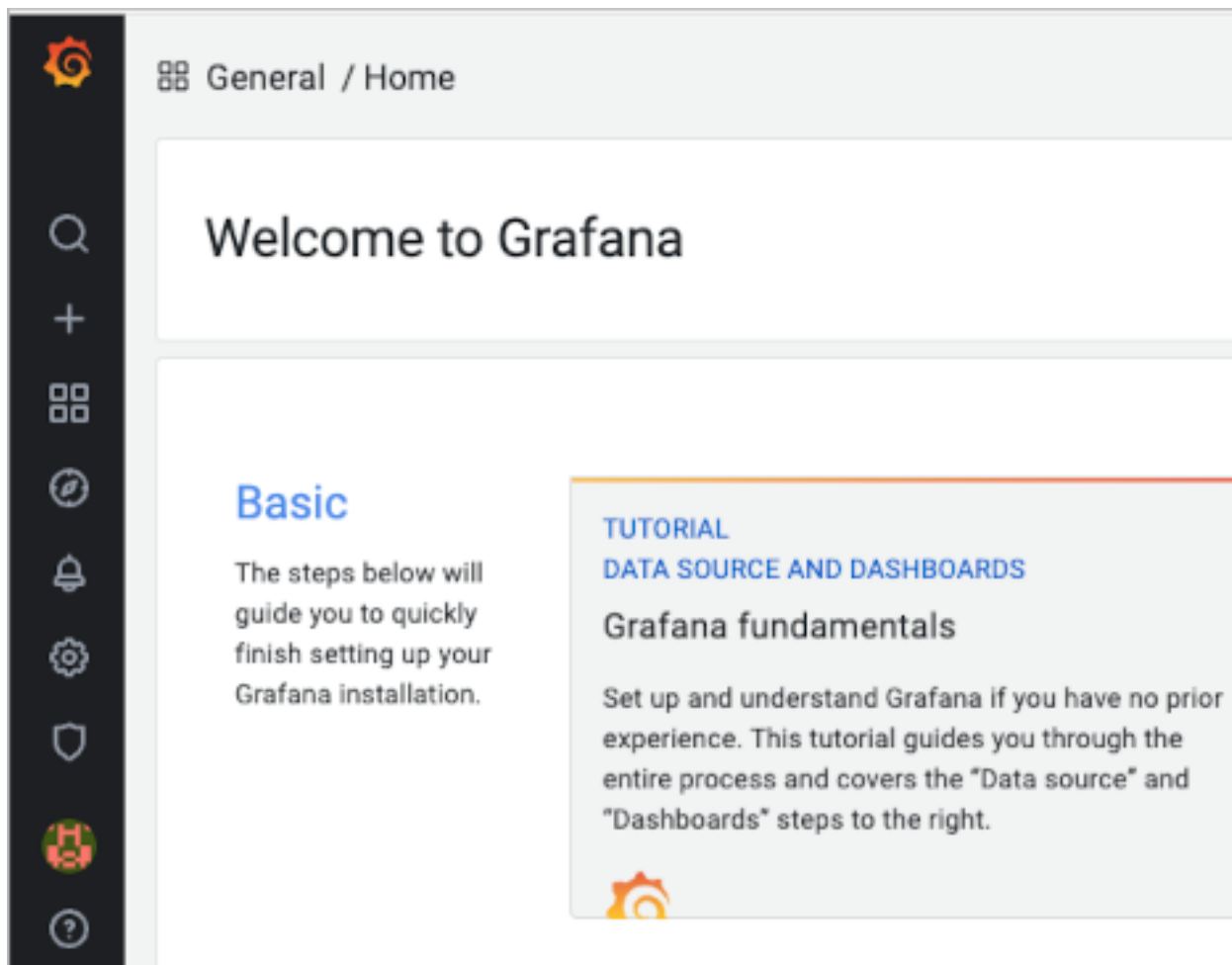
Contents

Monitoring resources with Grafana.....	4
Grafana in Cloudera Data Warehouse overview.....	4
Limitations of Grafana in Cloudera Data Warehouse.....	8
Getting started in Grafana.....	8
Viewing prebuilt dashboards.....	10
Monitoring HMS.....	14
Monitoring key Hive metrics.....	18
Monitoring Impala executors.....	21
Monitoring Impala catalog.....	24
Monitoring Impala statestore.....	26
Monitoring Impala admission control.....	27
Monitoring Impala coordinators.....	28
Monitoring nodes.....	29
Monitoring Kubernetes Services from Grafana.....	34
Updating a dashboard graph.....	38
Inspecting dashboard data and queries.....	42
Creating a custom dashboard.....	44
Meeting prerequisites to set up alerts.....	49
Creating alerts.....	57
Reviewing alerts and notifications.....	59
Forwarding Prometheus metrics from Cloudera Data Warehouse to an endpoint.....	62
Monitoring Kubernetes resources from K8s dashboard.....	63
Activating the K8S dashboard.....	63
Using the K8S dashboard.....	64
Forwarding logs to your observability system.....	68
Providing proxy CA certificates.....	73
OpenTelemetry support for Hive.....	73
Benefits of OTel Hive Integration.....	75
Configuring OTel in HiveServer2.....	75
Telemetry data exposed to OTel collector.....	76
Example Visualizations through OTel backend.....	79
Limitation of OpenTelemetry support for Hive.....	81

Monitoring resources with Grafana

Grafana is visualization and analytics software for using dashboards to monitor metrics data. You learn how to access pre-built Grafana dashboards to monitor Virtual Warehouses and your compute cluster in Cloudera Data Warehouse. As a Cluster Operations professional, you visualize Kubernetes and Istio metrics from Grafana to monitor and maintain the cluster.

When you log into Grafana, the Welcome page contains links to documentation about Grafana basics. From the left navigation, you can get started by searching for a particular dashboard or hovering over the navigation icons to familiarize yourself with the UI functions.



As a new user, to familiarize yourself with the pre-built dashboards that chart metrics about Hive, Impala, Hue, Druid, Kubernetes, and Istio in Cloudera Data Warehouse, follow steps in this documentation.

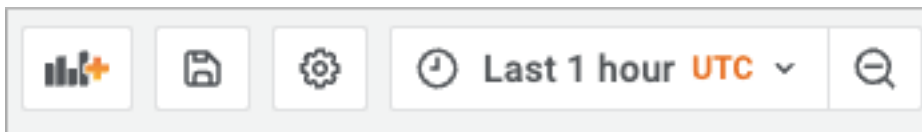
Related Information

[Grafana documentation](#)

Grafana in Cloudera Data Warehouse overview

You connect to prebuilt dashboards to view metrics of Cloudera Data Warehouse operations. Cloudera provides prebuilt Grafana dashboards for Hive, Impala, Hue, Druid, Kubernetes, and Istio dashboards of metrics data, charts, and other visuals.

Using Grafana, Cloudera metrics are centralized in a single spot, stored in the Prometheus database and monitored by Prometheus. Your workload databases are not involved in any way. You can immediately view pre-built dashboards described below. You can view dashboard metrics for different time periods by selecting the period of interest from the dropdown in the horizontal navigation:



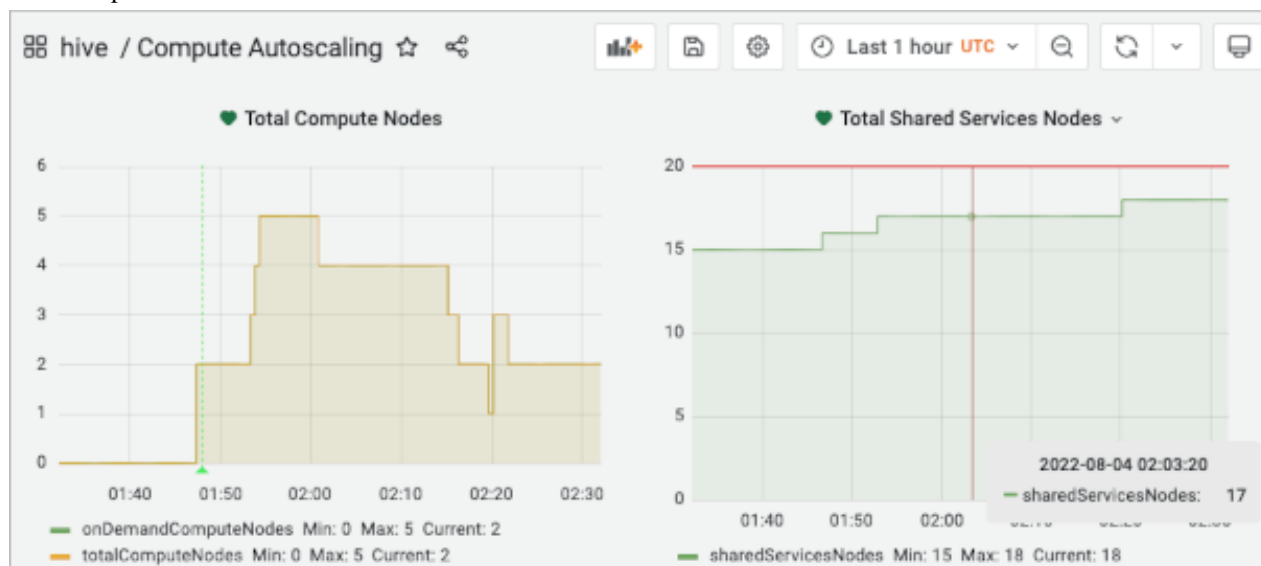
You can also create your own dashboards. "Get started with Grafana and Prometheus" describes how to create dashboards of Cloudera metrics. Describing all the details of how to use Grafana is beyond the scope of this documentation. Grafana described in this documentation is not the enterprise version.

Hive dashboards

The Hive dashboards cover the following operations of the Hive SQL engine in Cloudera Data Warehouse:

- Auto-scaling
- Hive metastore
- HiveServer
- The Hive service itself (Hive-Home)
- LLAP

For example:



Impala dashboards

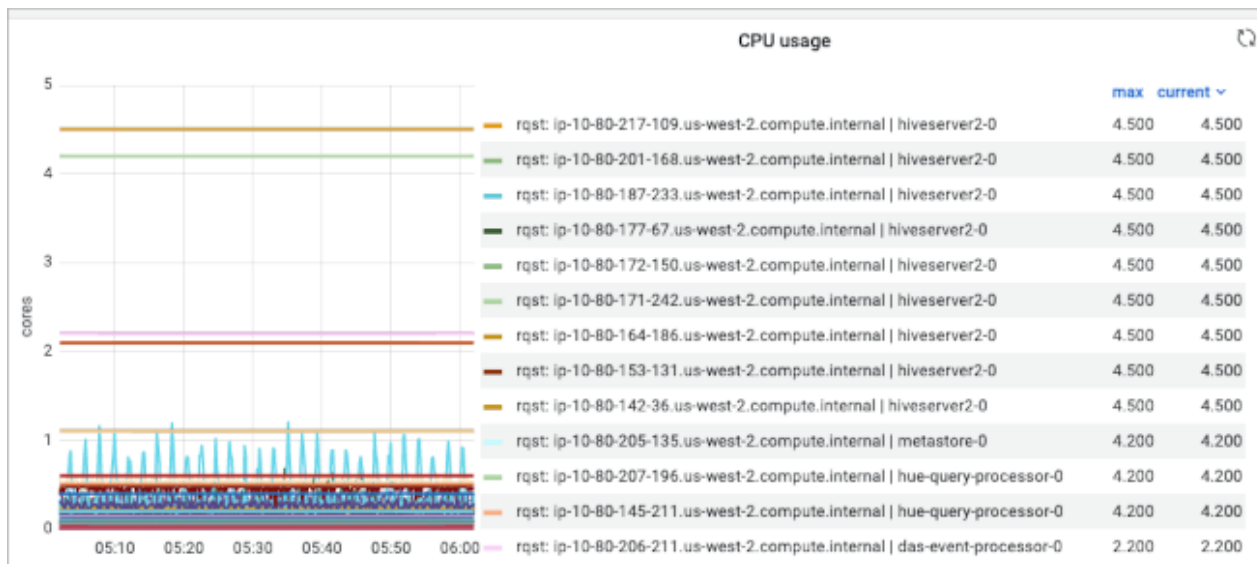
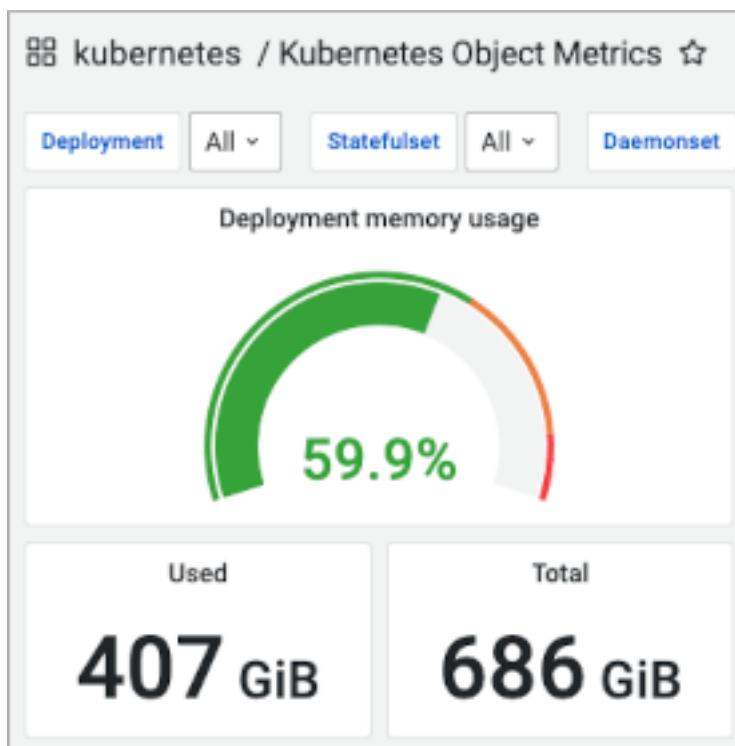
The Impala dashboards include the following operations of the Impala SQL engine in Cloudera Data Warehouse:

- Catalog server
- Coordinator
- Executor
- Statestore
- The Impala service itself

Kubernetes dashboards

You can get insight into the operations of your Cloudera clusters from Kubernetes dashboards. Kubernetes dashboards represent the following metrics:

- CoreDNS: requests and duration of responses
- App Metrics: number of Kubernetes pods, CPU usage relative to request or to limit, memory usage sliced and diced a number of ways
- Object Metrics: Deployment memory and CPU usage, in total, and by node.



Istio dashboards

To work with Cloudera Support, you might use the Istio dashboards. Istio is an open platform that provides microservice security, connections, and monitoring. The Istio Mesh dashboards cover the following views of the service mesh network of microservices:

- Istio Mesh summary: describes the network of microservices by HTTP/gRPC and TCP workloads in the Mesh.
- Individual Cloudera services, such as HiveServer and Impala coordinator: Request and response metrics, such as latency, for each mesh service (HTTP/gRPC and TCP) and client and service workloads metrics.

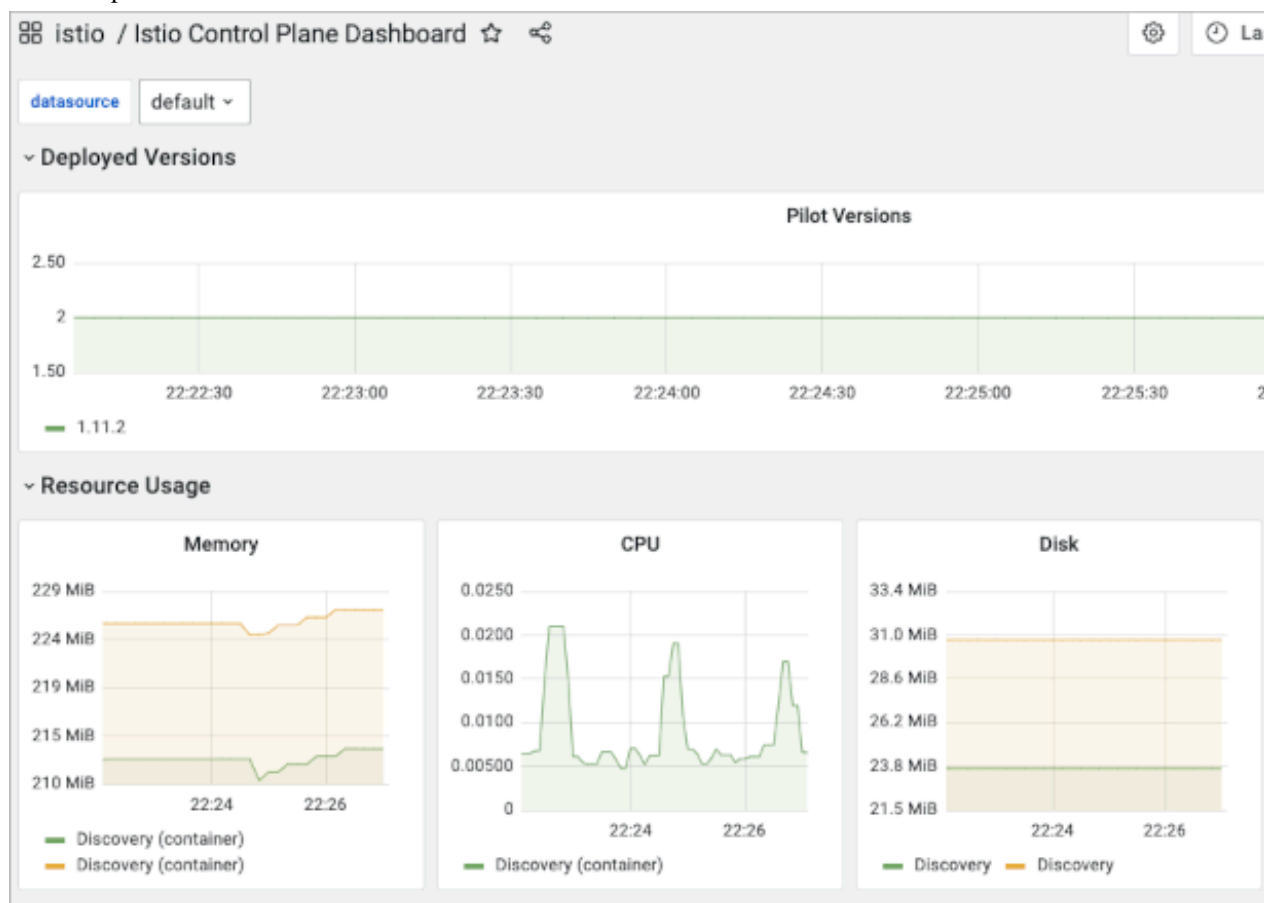
- Individual workloads using the services: Request and response metrics for each workload in the Mesh (HTTP/gRPC and TCP) and inbound/outbound workload services.

The Istio performance dashboard presents visualizations of the following metrics:

- Cloudera Usage
- Memory Usage
- Data Rates
- Bytes transferred per second

The Istio control plane dashboard includes memory, CPU, and disk resource usage.

For example:



The sidecar proxy metrics in Istio dashboards reveals the interceptions of network communication between microservices.

The Istio Wasm Plugin extends the Istio proxy capabilities.

Nodes dashboards

The [Prometheus Node Exporter](#) is used to gather detailed metrics for the AWS and Azure Virtual Machines that host an environment's Kubernetes cluster. These metrics cover both machine and OS level metrics such as CPU, memory, network, processes, time synchronization, disk, and file system. All metric names start with node_.

Three prebuild dashboards show metrics for AWS/Azure Virtual Machines that host the Kubernetes cluster:

- Cluster Totals

Shows CPU/Memory utilization and node counts (shared services/compute for the entire cluster).

- **Node Details**
Shows very detailed metrics for a single node at a time.
- **Node Trends**
Combines CPU, Memory, Disk, and Network metrics for all nodes for node-to-node comparisons.

Related Information

[Grafana documentation](#)

[Get started with Grafana and Prometheus](#)

[Istio documentation](#)

[Istio in-depth documentation](#)

[Istio github site](#)

[Wasm Plugin](#)

Limitations of Grafana in Cloudera Data Warehouse

You need to understand the Grafana capabilities in Cloudera Data Warehouse that Cloudera does not support. Grafana in Cloudera Data Warehouse is intended for use by cluster operations professionals who are familiar with monitoring tools, interpreting metrics, and performing maintenance.

Unsupported features

Storing metrics longer than 15 days or consuming more than 90GB of disk space is not supported. Metrics older than 15 days will be deleted. If the stored metrics consume more than 90GB of disk space, metrics will be deleted regardless of the number of days stored.

Custom dashboards you create in Grafana are lost upon restarting or updating the cluster. Only the default dashboards are supported.

Getting started in Grafana


Learn how to log into Grafana in Cloudera Data Warehouse on cloud.

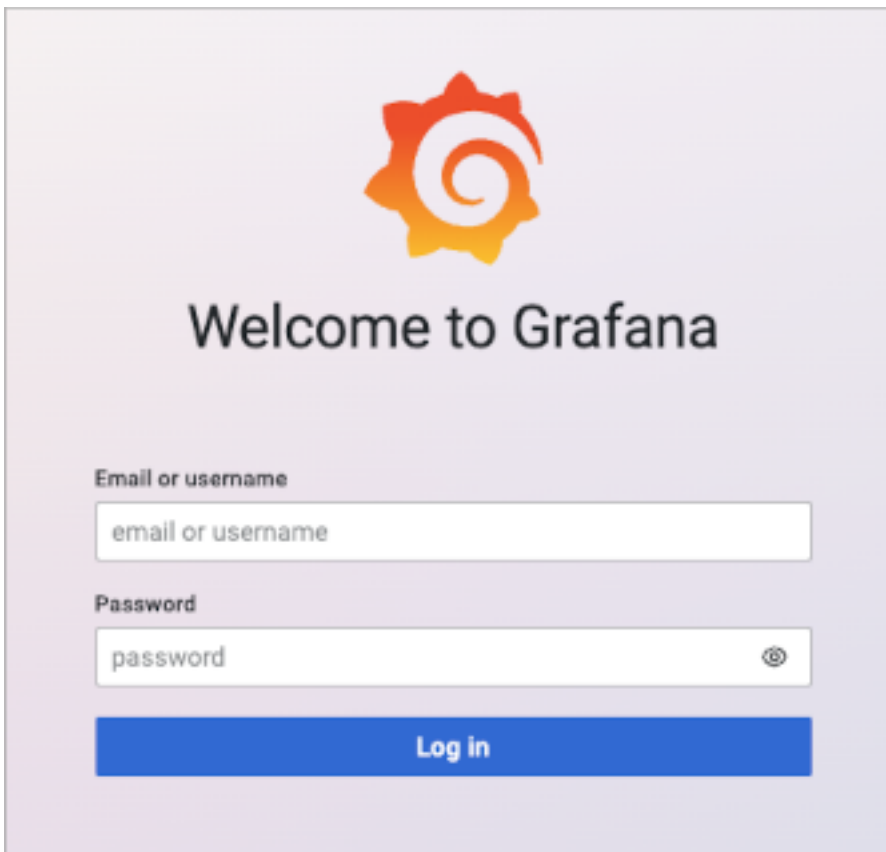
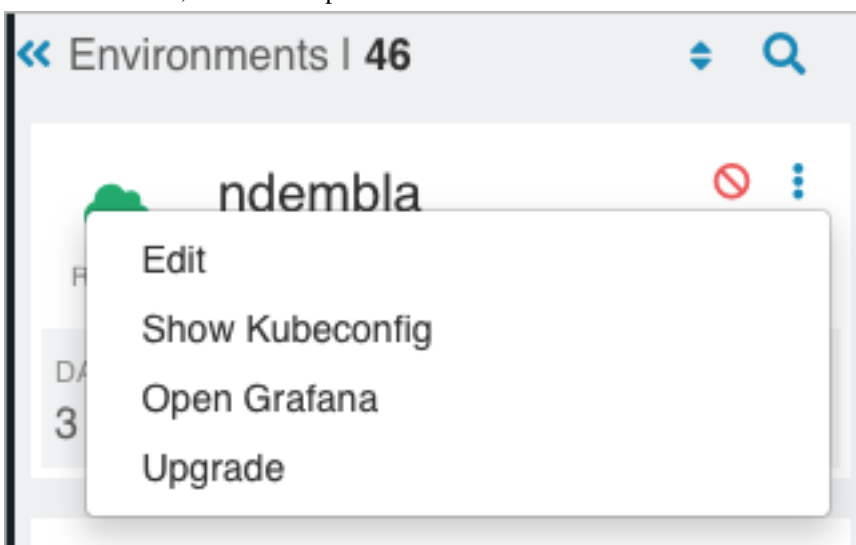
Before you begin

- To access Grafana, your Cloudera administrator must use Cloudera Management Console to assign the environment to you as a resource.
- You must obtain the DWAdmin role.

Procedure

1. In Data Warehouse Overview , expand the Environment column.

2. Search for the environment that is associated with the Virtual Warehouse, click more options  in the environment tile, and select Open Grafana.



3. Enter your LDAP user name, or the name provided by your Administrator for logging into Grafana.
4. Go back to the environment tile, and select Copy Kubeconf.
5. Open a terminal window, paste the contents of your clipboard into a text file, and save the file by the name kubeconf (no extension).
6. Export the KUBECONFIG variable equal to the file name.

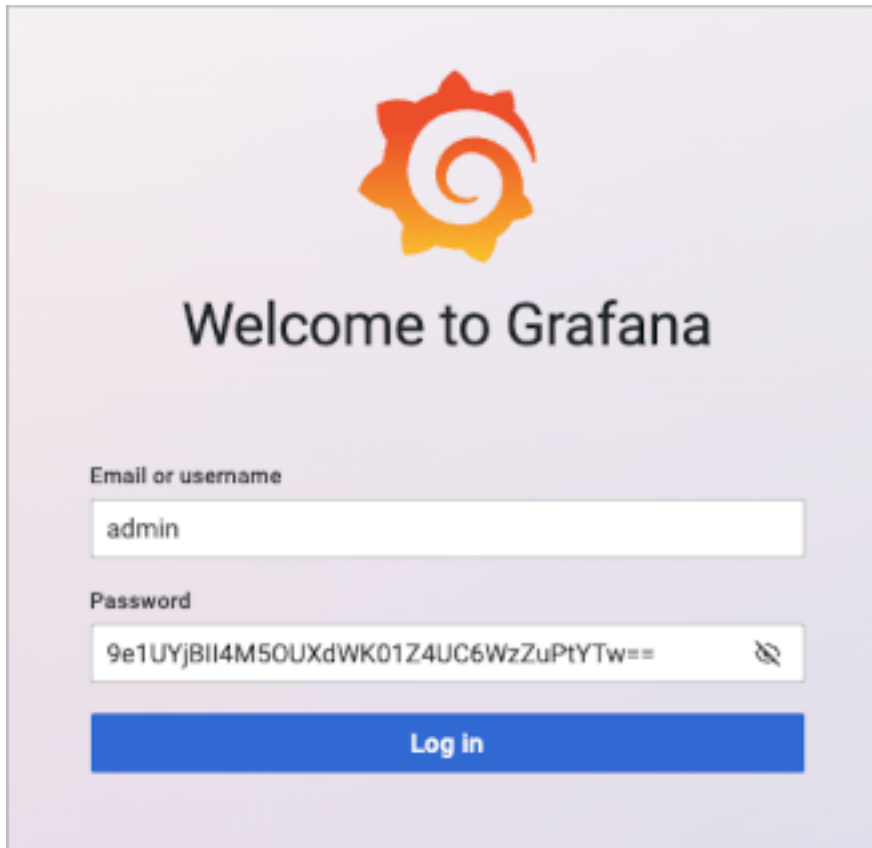
```
export KUBECONFIG=kubeconf
```

7. Get the password for Grafana.

For example, use the `kubectl` command as follows:

```
kubectl get secret grafana -n istio-system -o jsonpath="{.data.passphrase}" | base64 -D | pbcopy
```

8. Go back to the Grafana login dialog, and paste the contents of your clipboard.
The Grafana login dialog should look something like this:

The image shows the Grafana login interface. At the top is the Grafana logo, a stylized orange and yellow gear with a white spiral. Below the logo, the text "Welcome to Grafana" is displayed in a large, dark font. Underneath, there are two input fields. The first is labeled "Email or username" and contains the text "admin". The second is labeled "Password" and contains a long, base64-encoded string: "9e1UYjBII4M5OUXdWK01Z4UC6WzZuPtYTW==". To the right of the password field is a small icon of a crossed-out eye, indicating that the password is hidden. Below the input fields is a large blue button with the text "Log in" in white.

9. Click Log in.

Related Information

Viewing prebuilt dashboards


You see how to list dashboard groups, view the Hive dashboard, and see the actual metric data on a point in the X-axis. You can follow the same steps to work with other dashboards.


Procedure

1. Log into Grafana as described in the previous topic, "Getting Started in Grafana".

2.





In the Welcome screen, click grid , and then select Manage.
A list of dashboard groups appears:





Dashboards

Manage dashboards and folders


 **Browse**

 Playlists



 Snapshots


 Library panels









New ▾

 Filter by tag ▾

☐ Starred

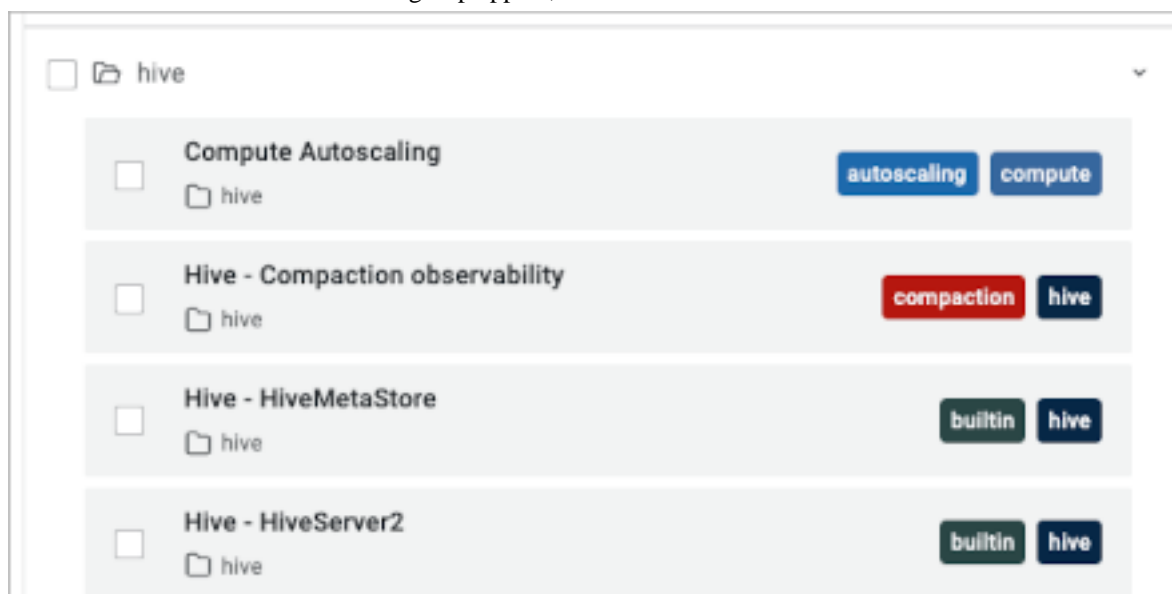
 Sort (Default A-Z) ▾

<input type="checkbox"/>  General	▾
<input type="checkbox"/>  druid	▾
<input type="checkbox"/>  hive	▾
<input type="checkbox"/>  hue	▾
<input type="checkbox"/>  impala	▾
<input type="checkbox"/>  istio	▾
<input type="checkbox"/>  kubernetes	▾
<input type="checkbox"/>  nodes	▾

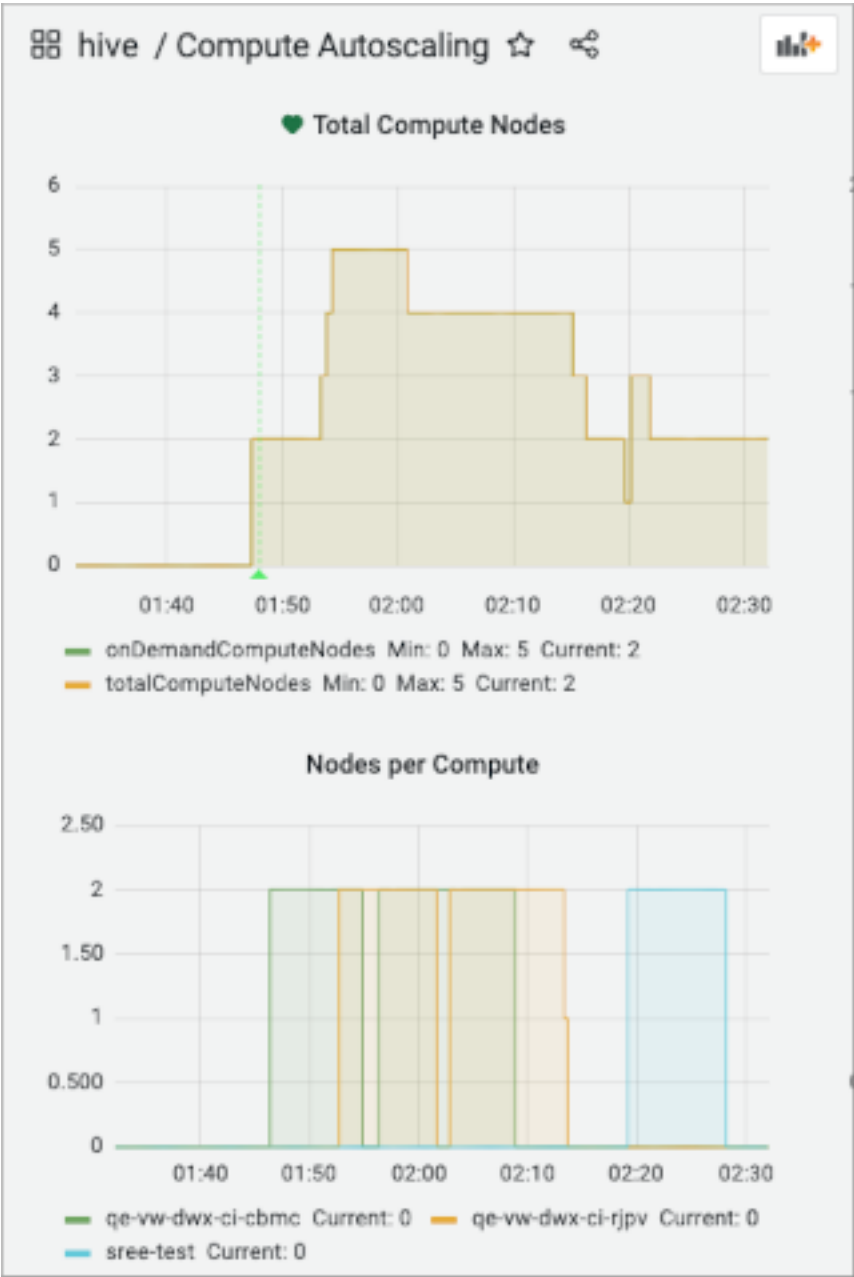
3. Click the name of a dashboard group.

For example, click hive.

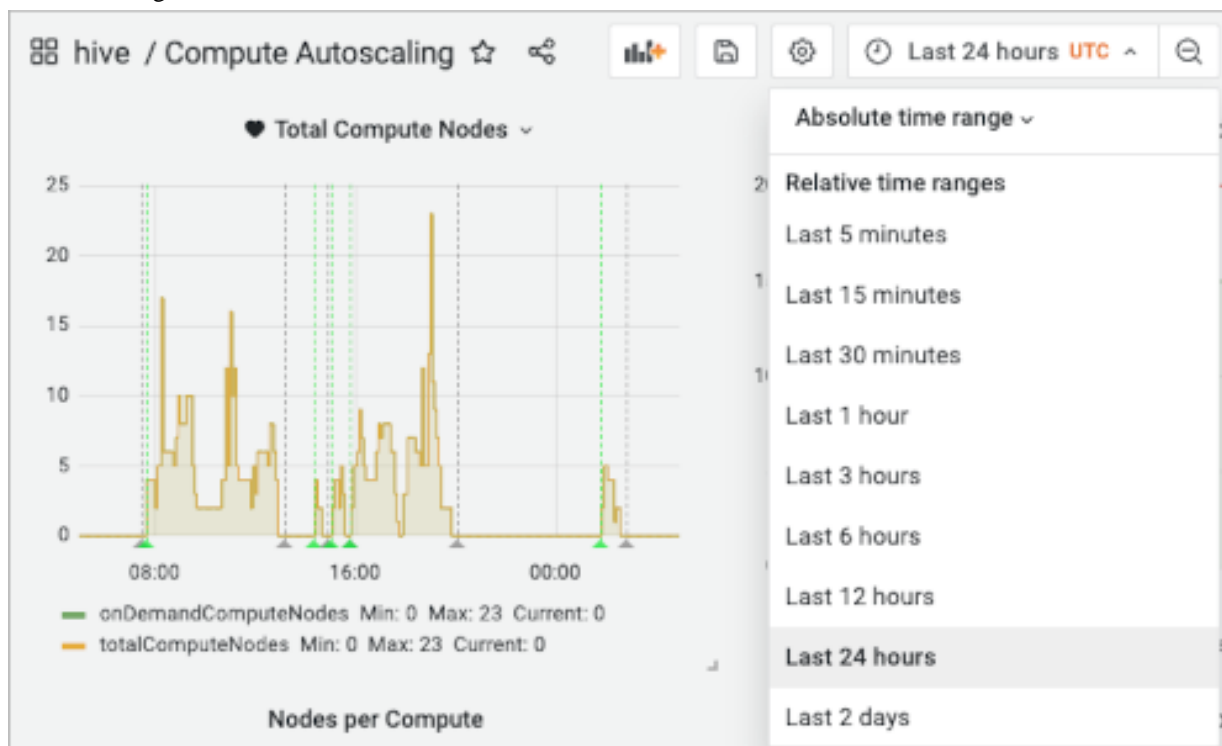
Names of the Hive dashboards in the group appear, a few of which are shown below:



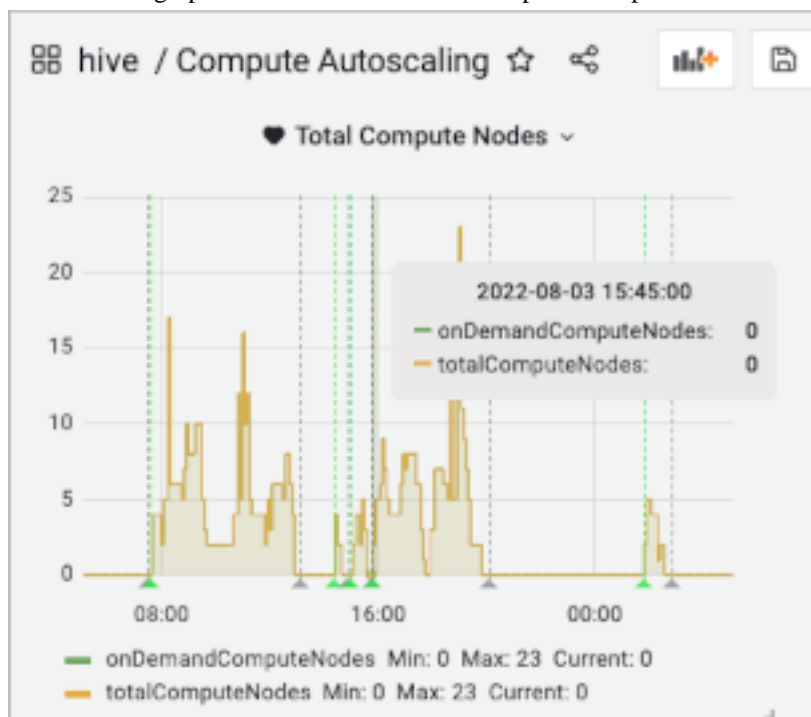
4. Click the name of a dashboard group.
For example, click Compute Autoscaling.
A number of dashboards appear. Only two are shown below:



5. Select viewing of metrics over the last 24 hours.



6. Hover over a graphic to reveal metric data for a particular point on the X-axis.




Related Information

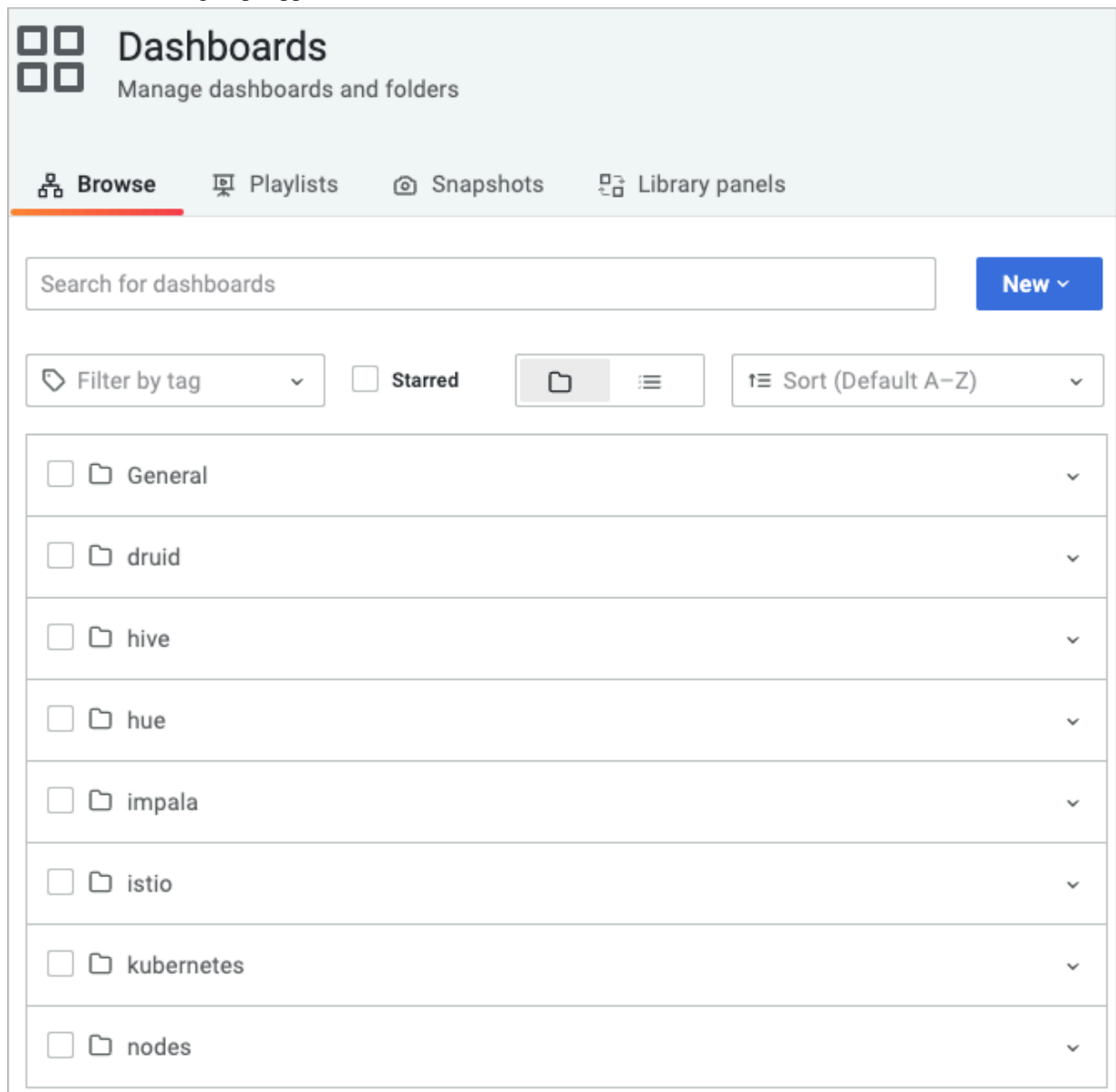
Monitoring HMS

You can monitor Hive Metastore (HMS), heap usage, and key Hive metrics.

Procedure

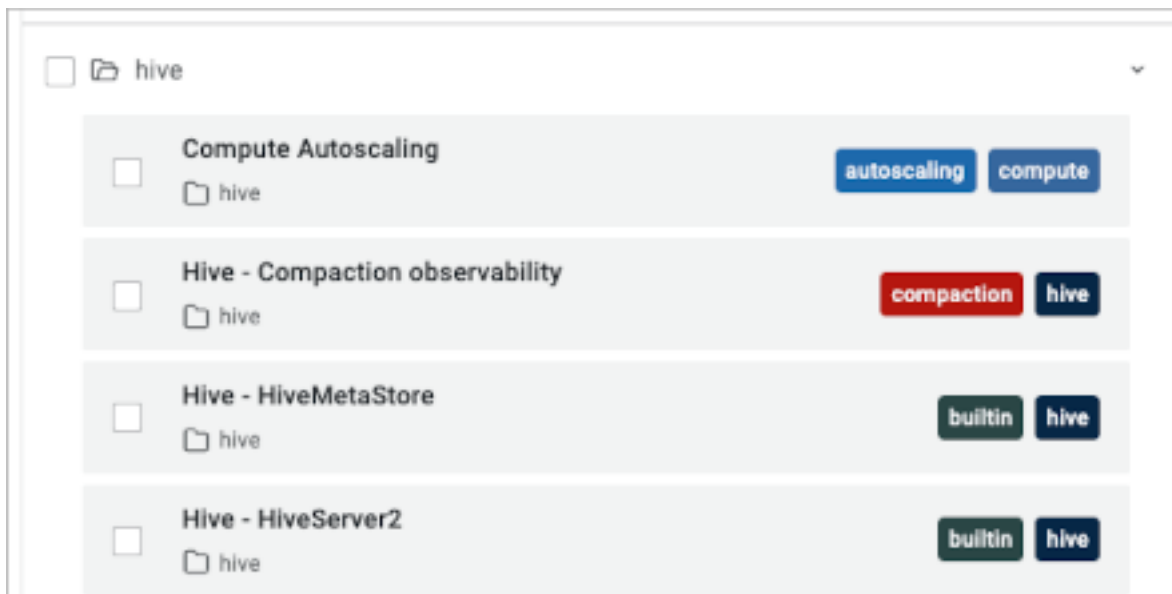
1.

In the Welcome screen, click grid , and then select Manage. A list of dashboard groups appears:



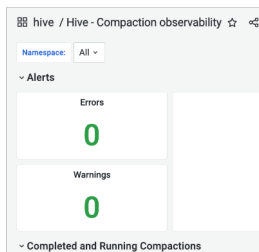
2. Select the hive dashboard.

Names of the Hive dashboards in the group appears, a few of which are shown below:



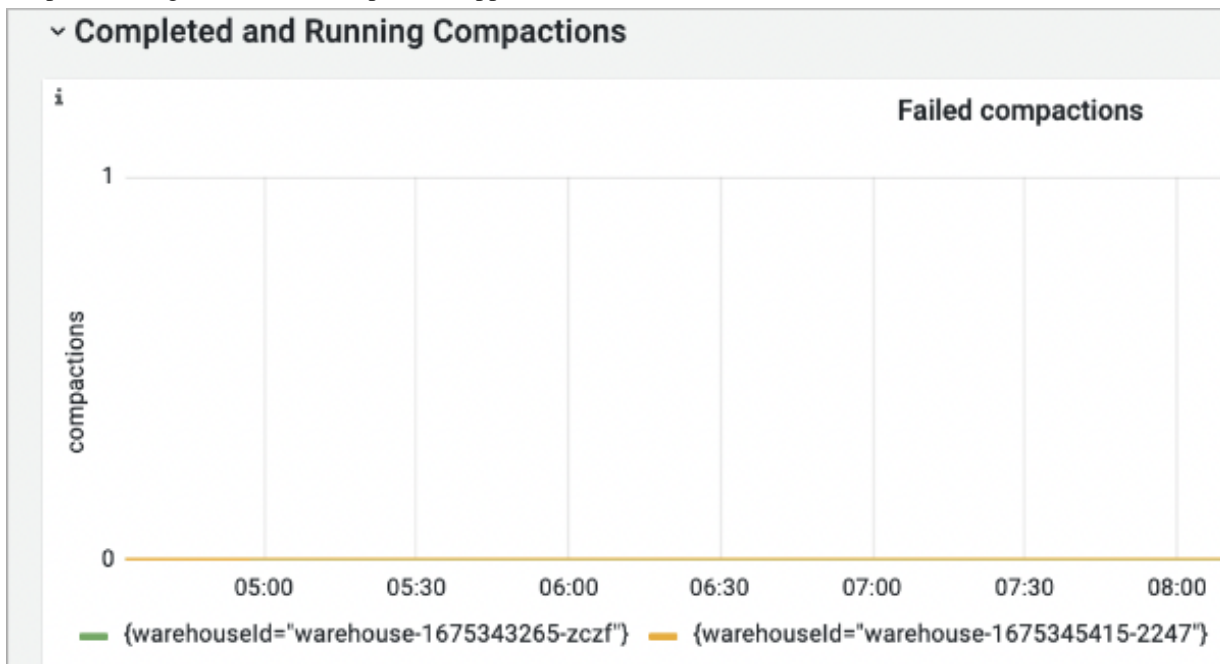
3. Select Hive - Compaction observability, and in Namespace, select a namespace.

In Alerts, the error indicator shows the number of errors and warnings.

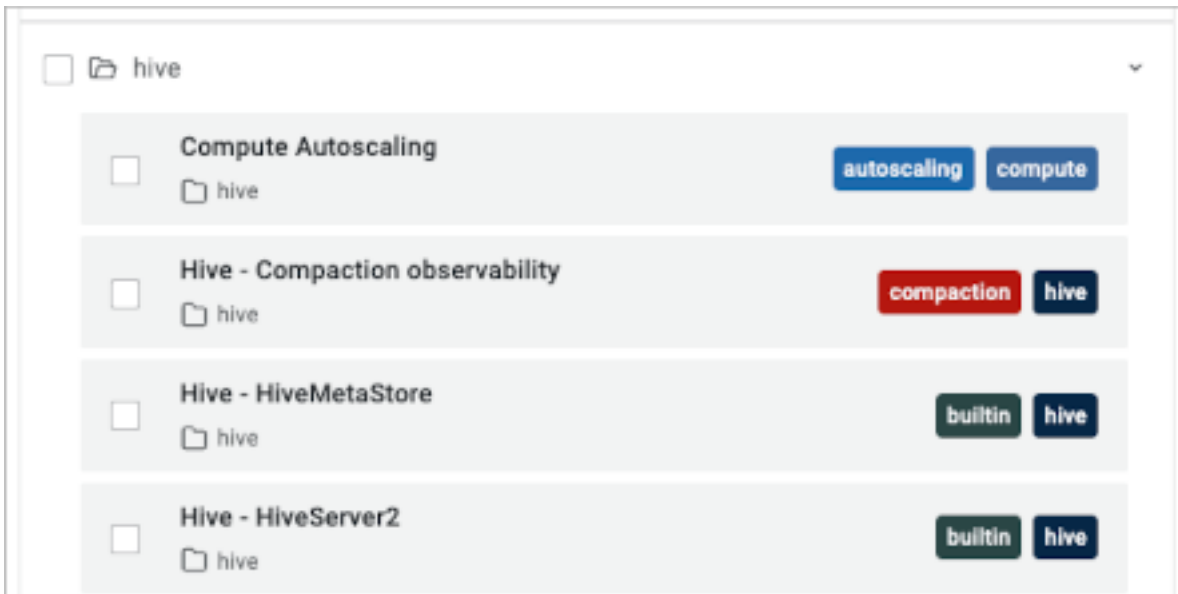


4. Expand Completed and Running Compactions.

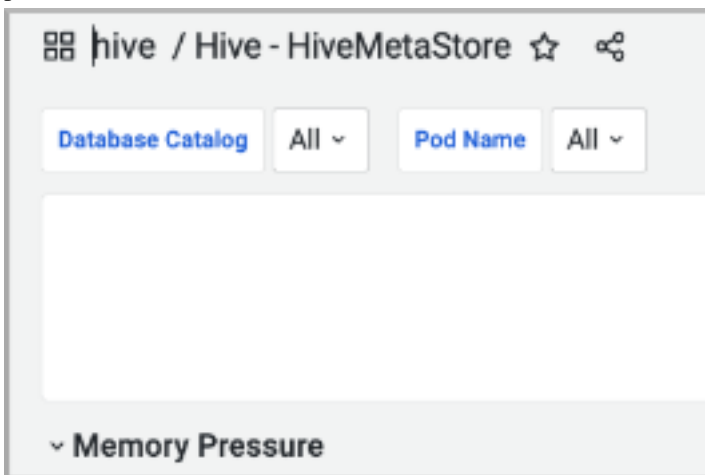
Graphs showing the status of compactions appears.



5. Go back to the list of Hive dashboards, and select Hive - HiveMetaStore.

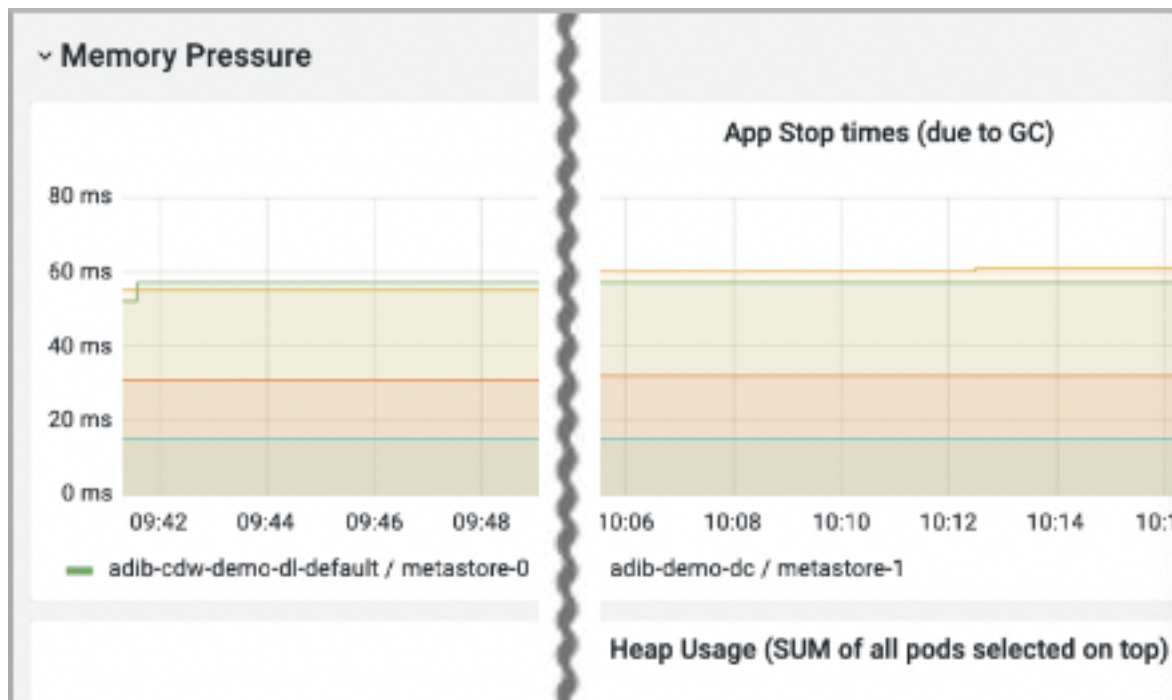


6. In Database Catalog, and then in Pod Name, accept the default All, or select a particular Database Catalog and pod.



In Memory Pressure, metrics about the overall status of HMS appear.

- Click each row title to expand and look at various metrics: app stop times due to garbage collection and heap usage.

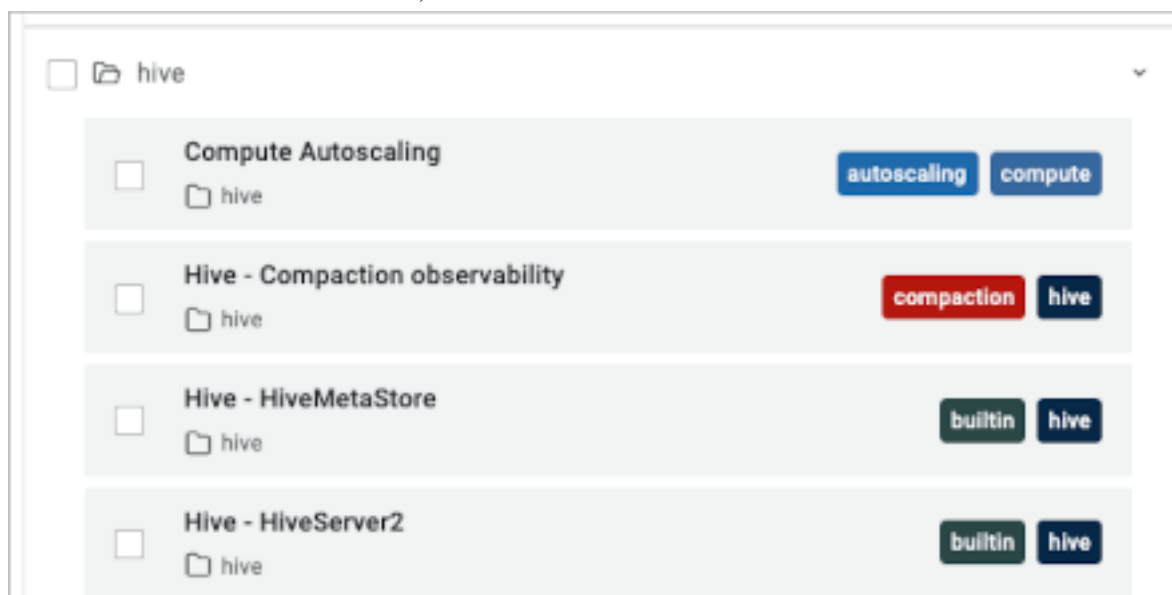


Monitoring key Hive metrics

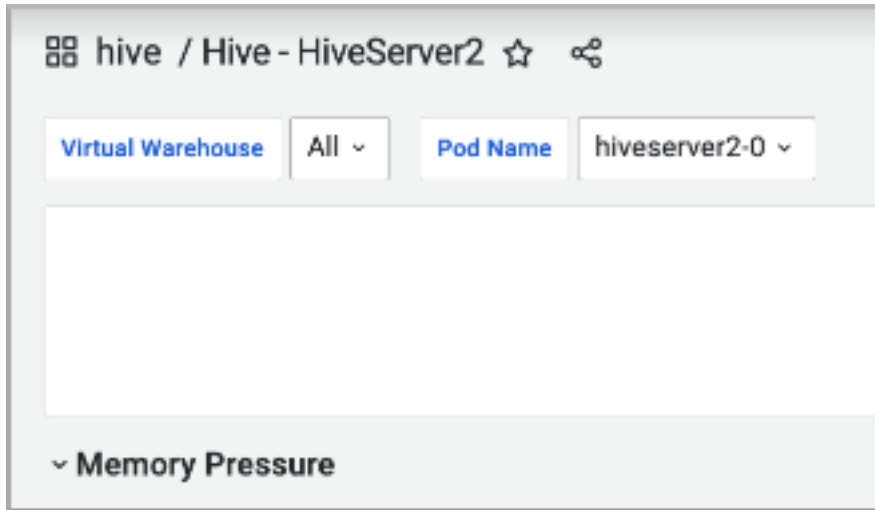
You can get metrics about the status of HiveServer.

Procedure

- Go back to the list of Hive dashboards, and select Hive - HiveServer2.

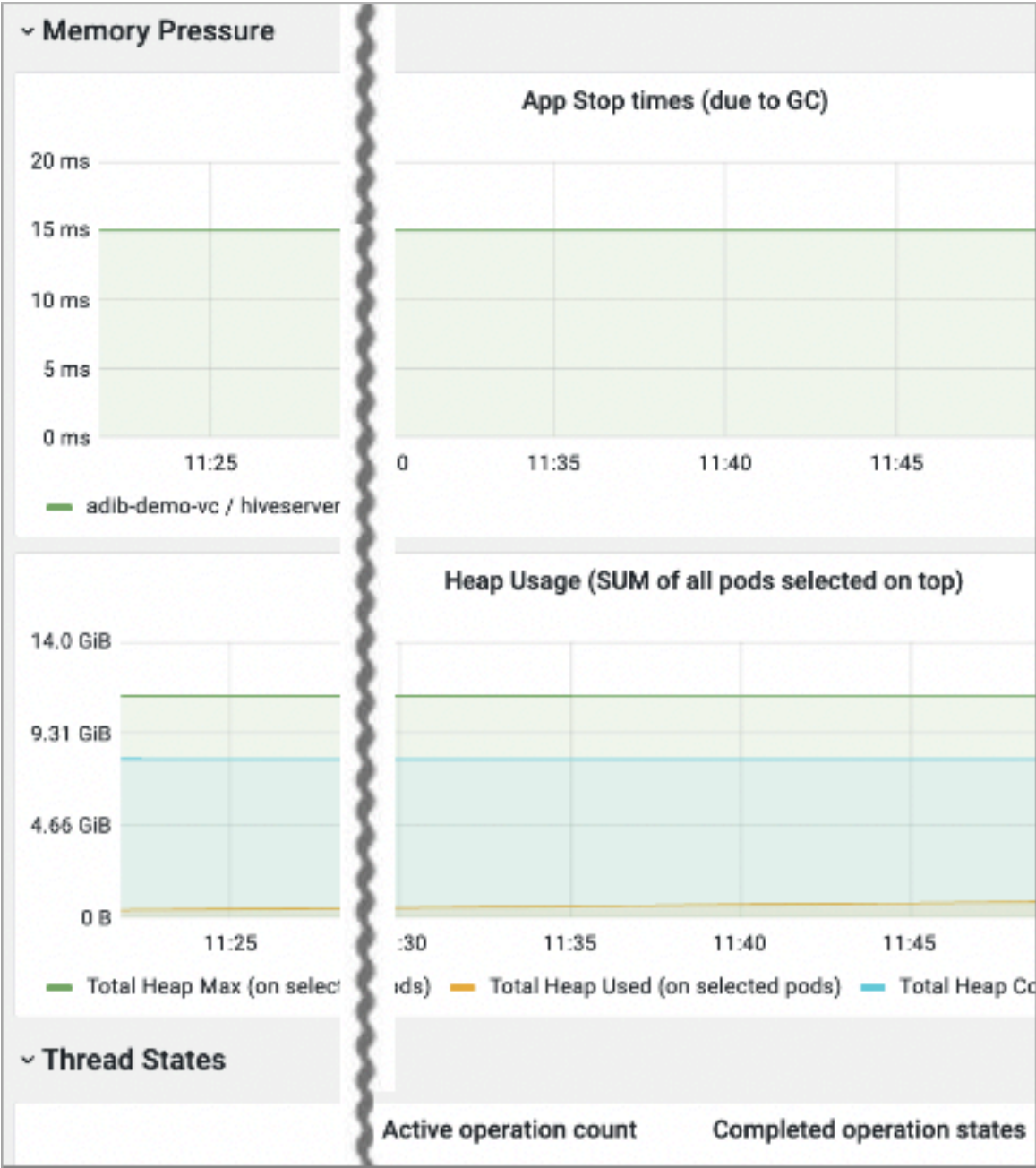


2. In Virtual Warehouse, and then in Pod Name, accept the default All, or select a particular Virtual Warehouse and pod.



In Memory Pressure, metrics about the overall status of HiveServer memory usage appear.

3. Click each row title to expand and look at various metrics: App stop times due to garbage collection, heap usage, active thread operation count, and completed thread operation states.



4. In hive / Hive - Home, you can monitor warehouse growth from metics on the tables and partitions created hourly.




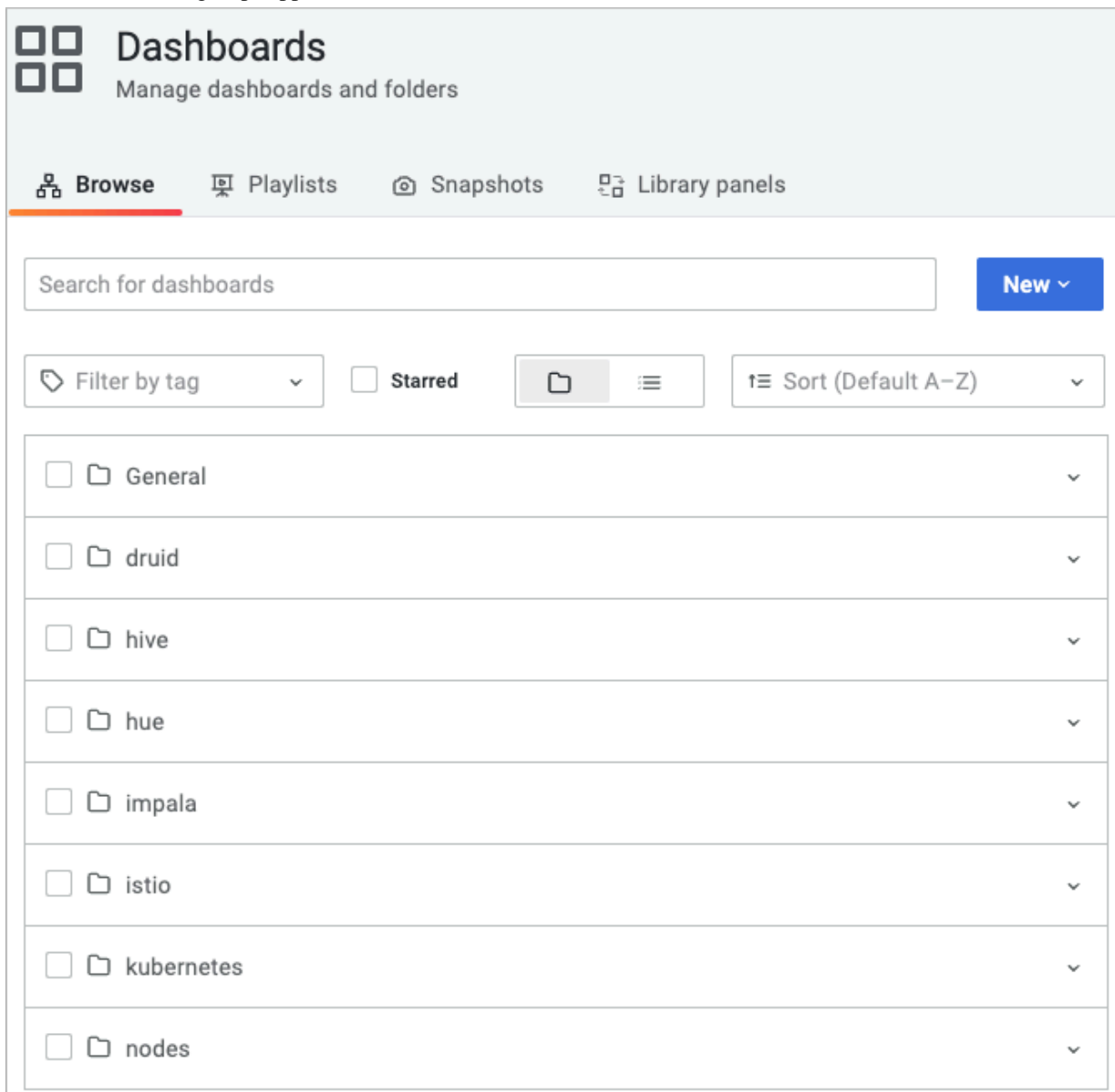
Monitoring Impala executors

You can monitor Impala executors and get related metrics.

Procedure

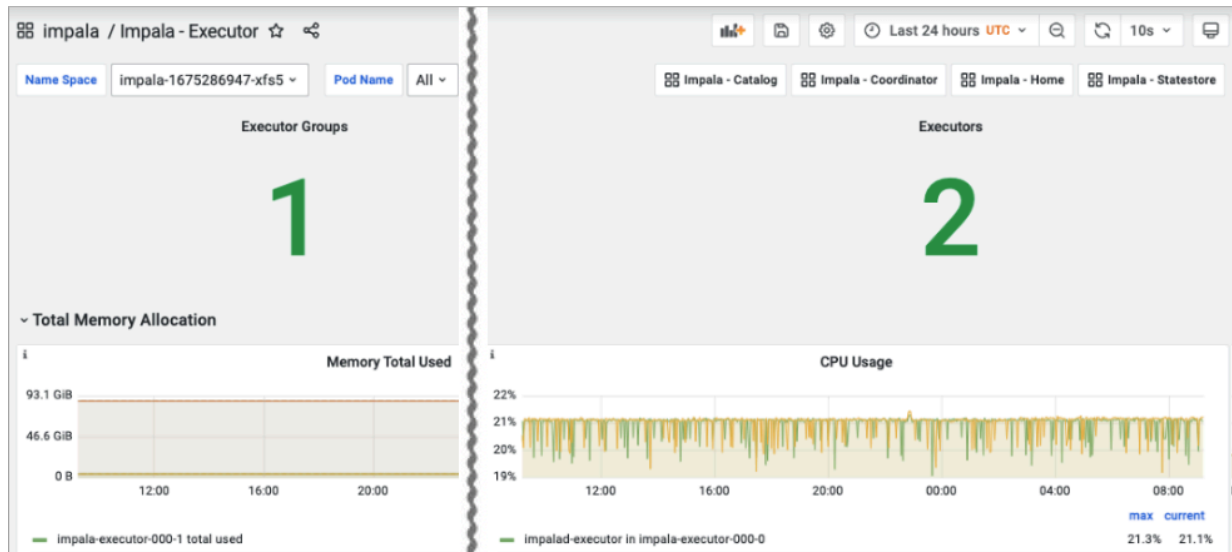
1.

In the Welcome screen, click grid , and then select Manage.
A list of dashboard groups appears:

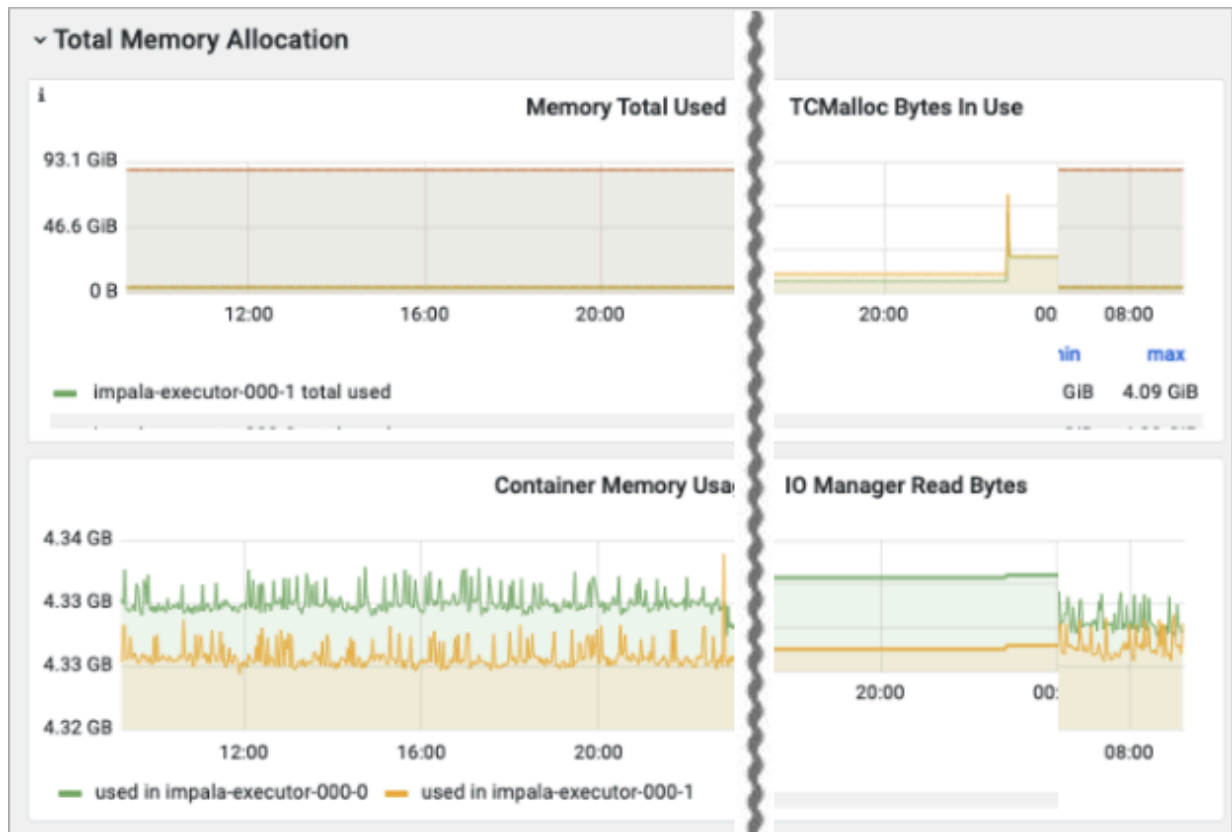


2. Click the impala dashboard group.

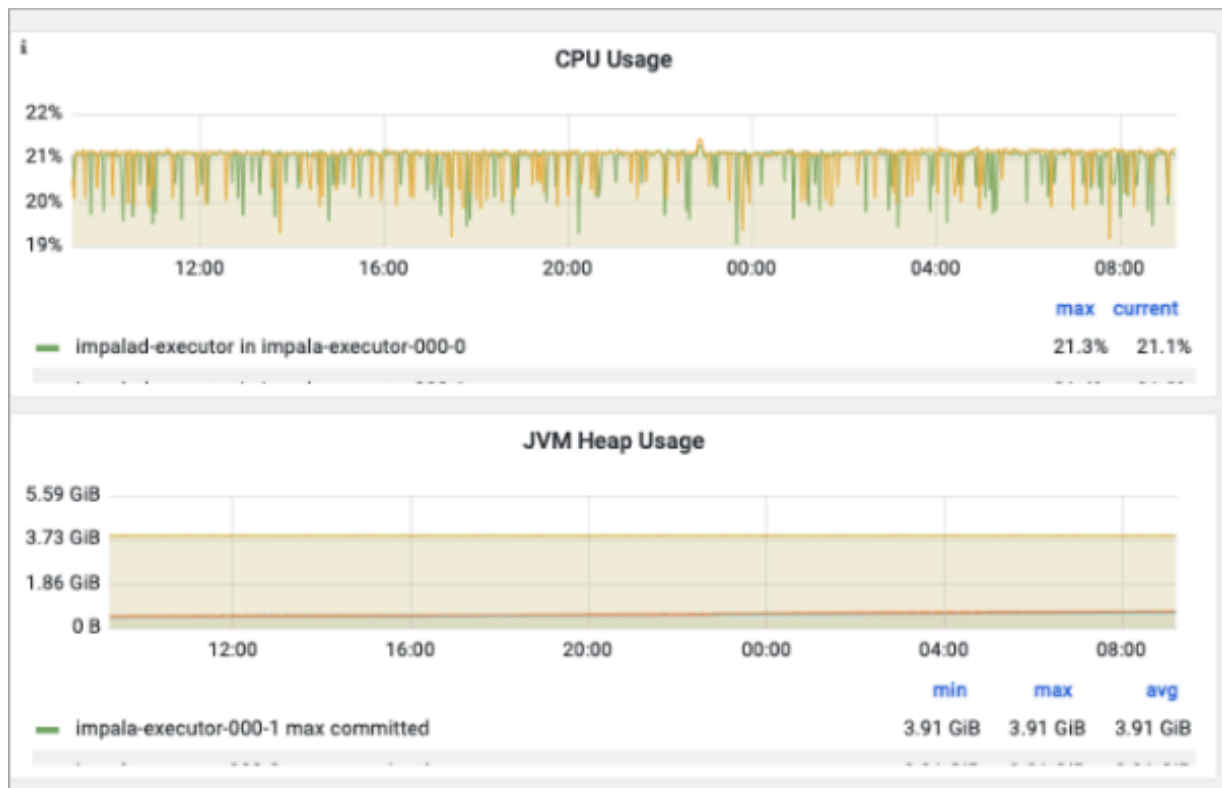
3. In the list of Impala dashboards, and select Impala - Executor.
The number of Executor Groups and Executors appear.



4. View graphs and metrics of memory usage.



5. Get information about Executor node CPU usage.

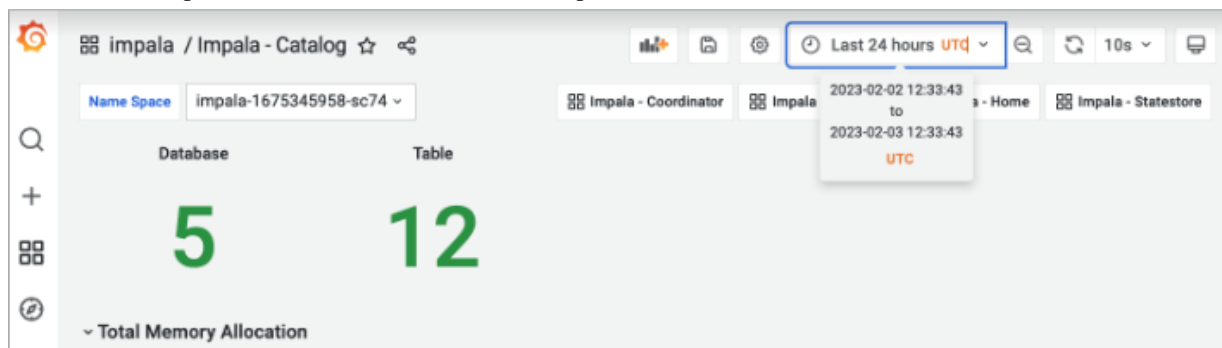


Monitoring Impala catalog

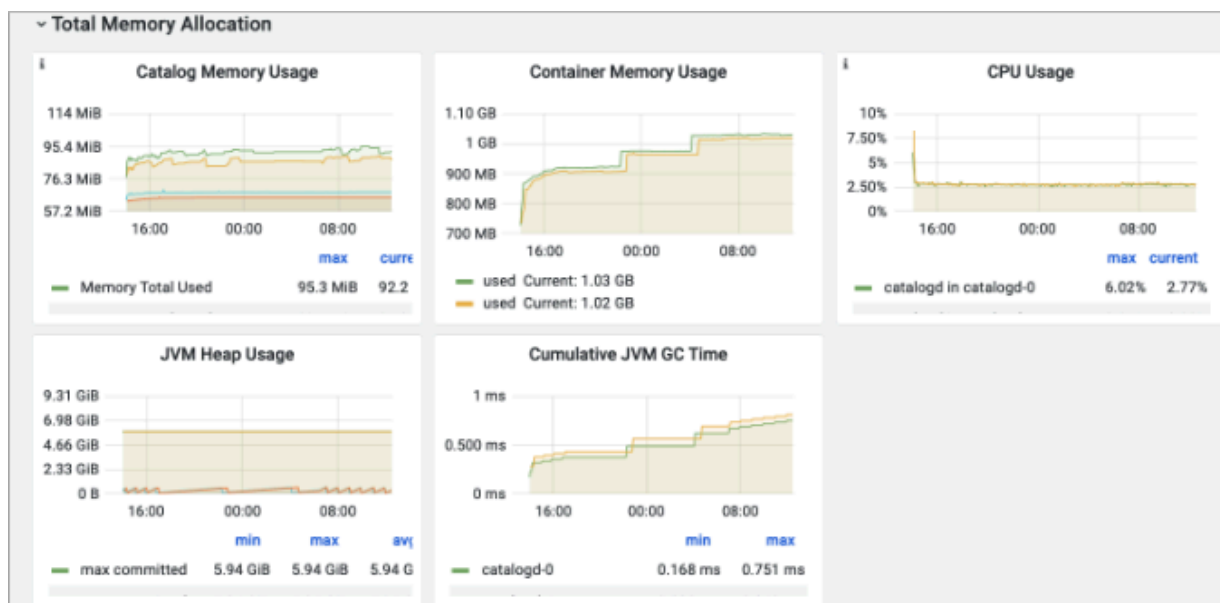
You can monitor Impala catalogd from Grafana.

Procedure

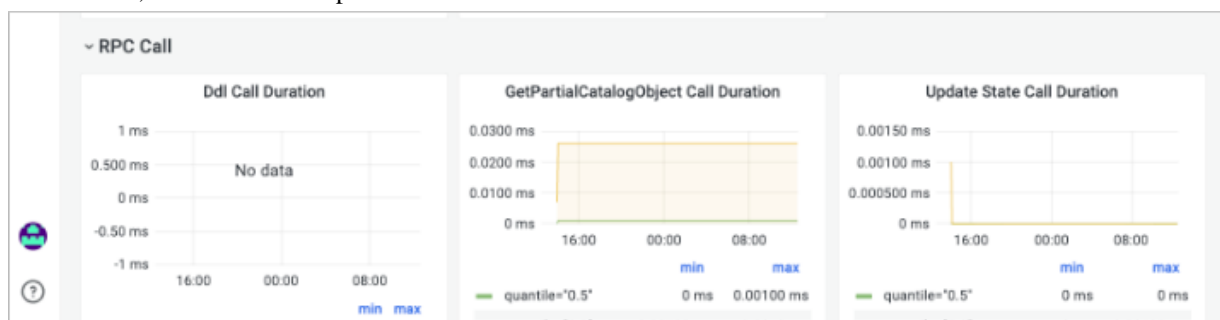
1. In the list of Impala dashboards, and select Impala - Catalog.
The number of databases and tables appear.
2. Select a Name Space, and from the Last 24 hours dropdown, select an time interval to monitor.



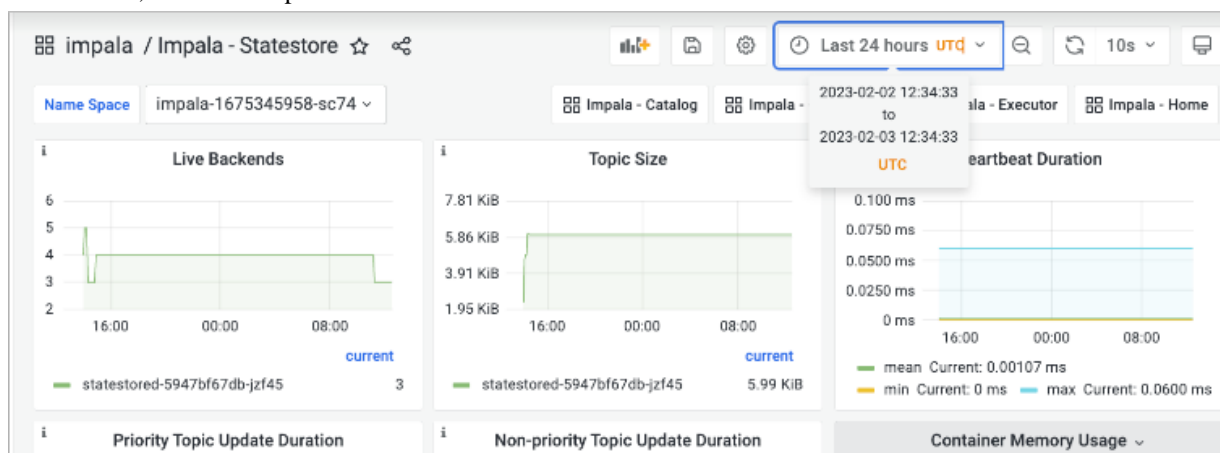
- View catalog memory usage, container memory usage, cpu usage, JVM heap usage, and cumulative JVM garbage collection time.



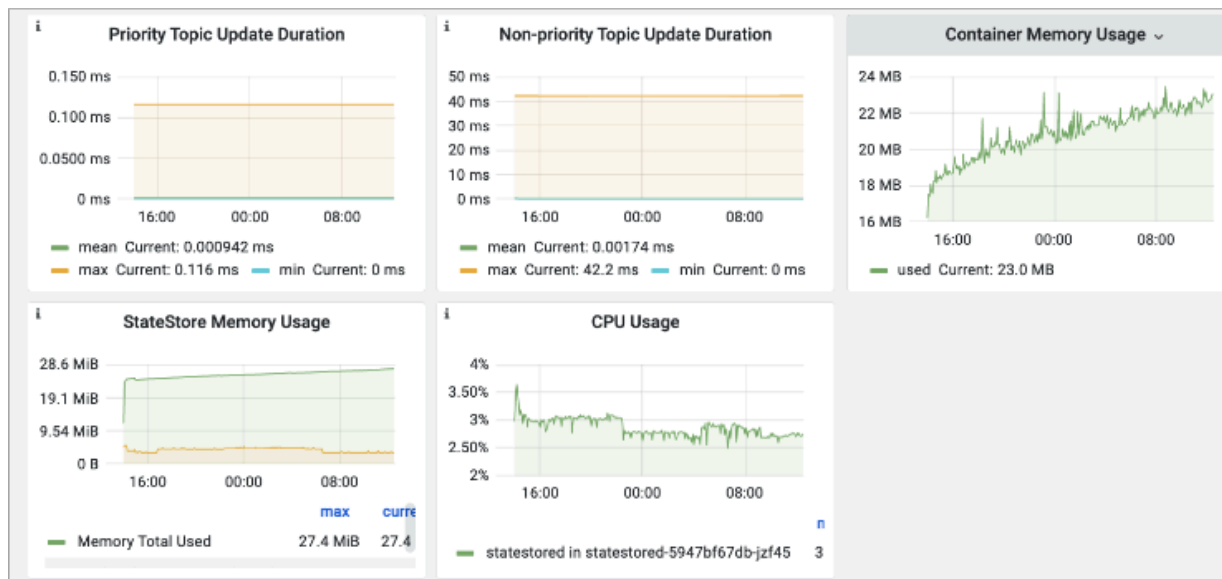
- Scroll down, and view remote procedure call metrics.



- Scroll down, and view Impala statestore metrics.



6. Scroll down, and view update, memory, and CPU metrics.

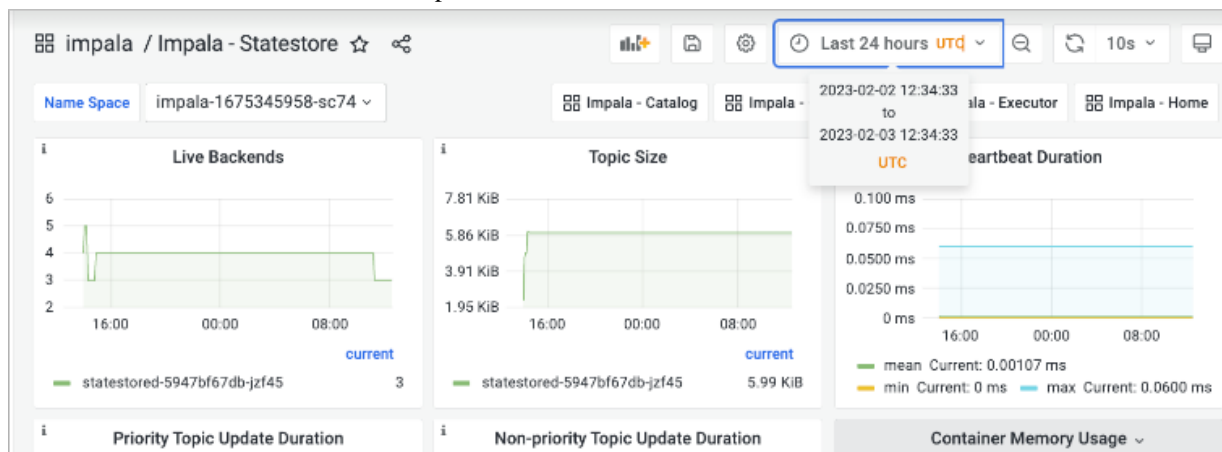


Monitoring Impala statestore

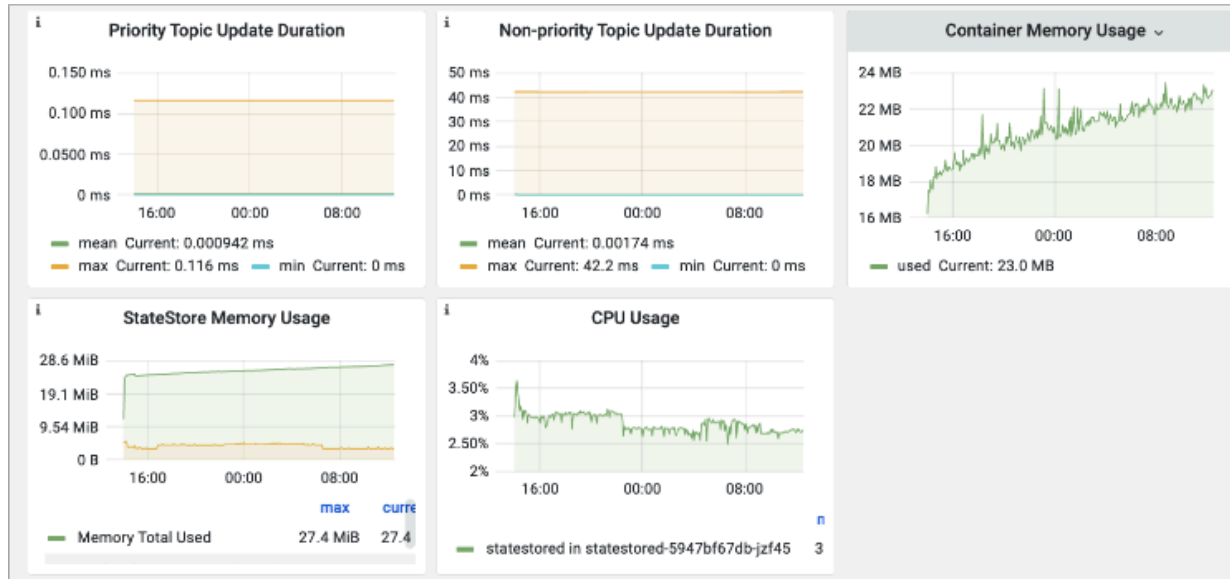
You can monitor Impala statestore from Grafana.

Procedure

1. In the list of Impala dashboards, and select Impala - Statestore.
2. Scroll down, and view live backends, topic size, and heartbeat duration.



3. Scroll down, and view priority and non-priority topic update duration, container memory usage, statestore memory, and cpu usage.

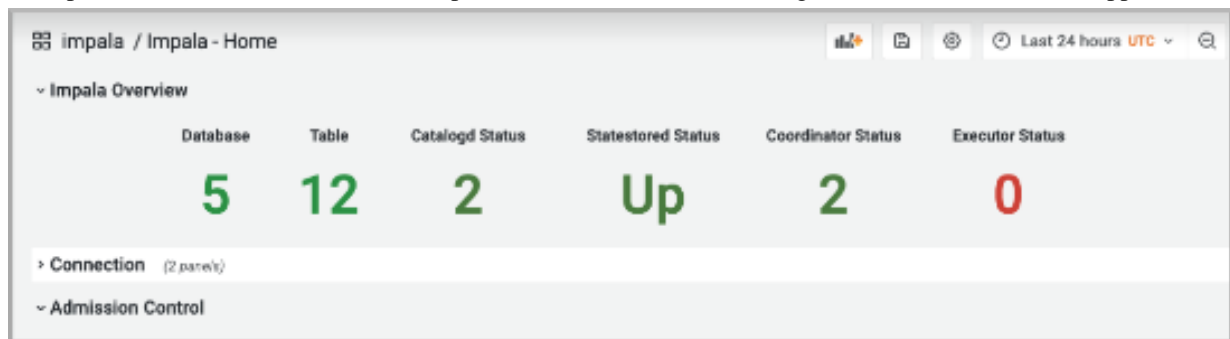


Monitoring Impala admission control

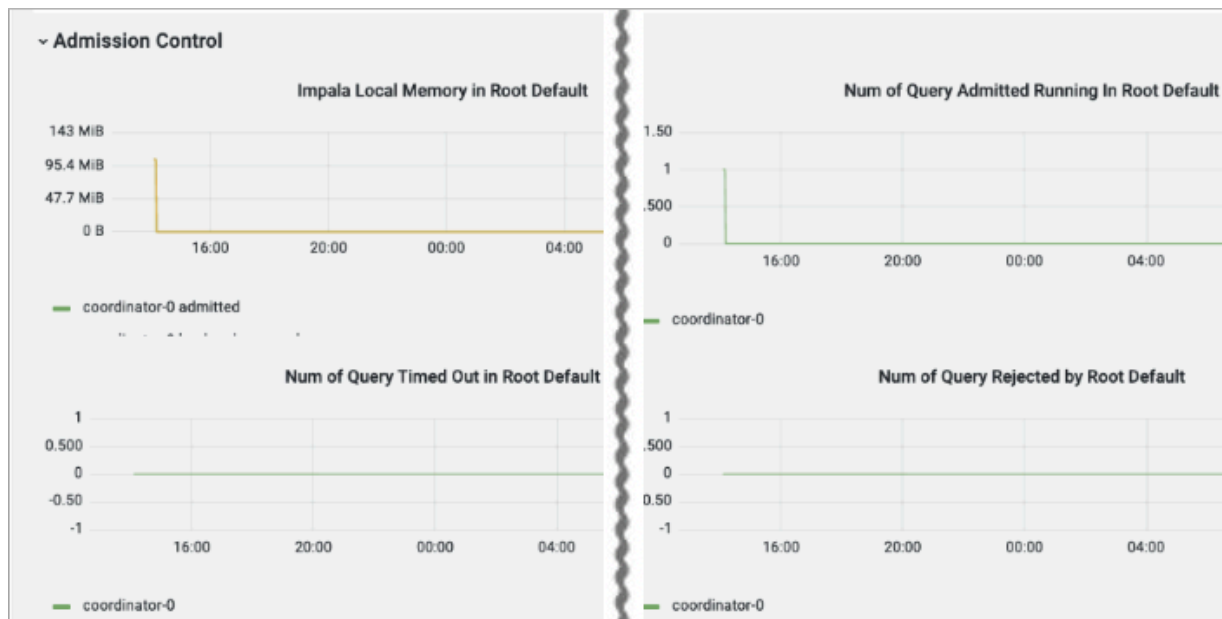
You can monitor Impala admission control from Grafana.

Procedure

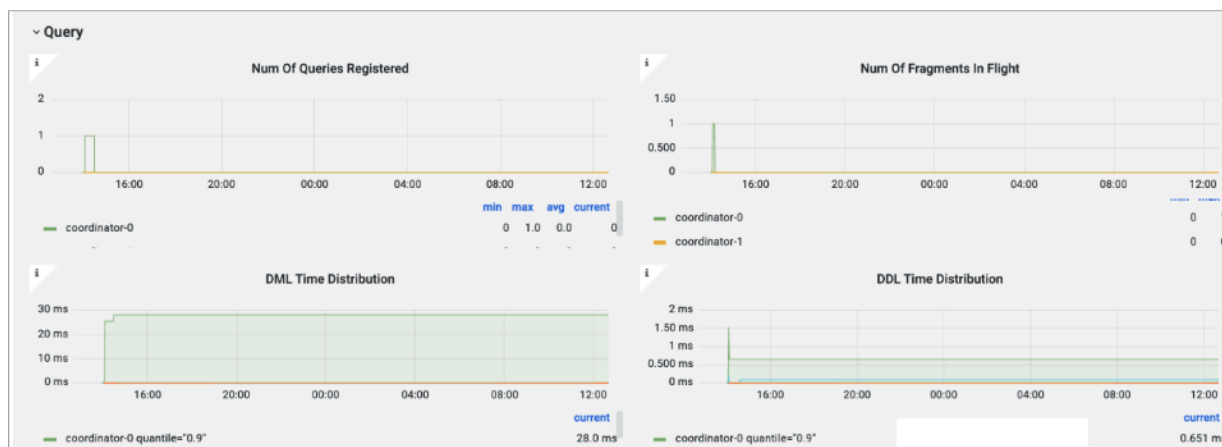
1. In the list of Impala dashboards, and select Impala - Home.
In Impala Overview, the number of of Impala databases, tables, the catalogd status, and other metrics appear.



2. In Admission Control, view the Impala local memory in root default, the number of queries admitted that are running in root default, and the queries in root default (not shown).



3. View the number of queries timed out and rejected in root default.
4. Scroll down and in Query view the number of registered queries, fragments in flight, number of open files (not shown), data manipulation language (DML) time distribution, and data definition language (DDL) time distribution.



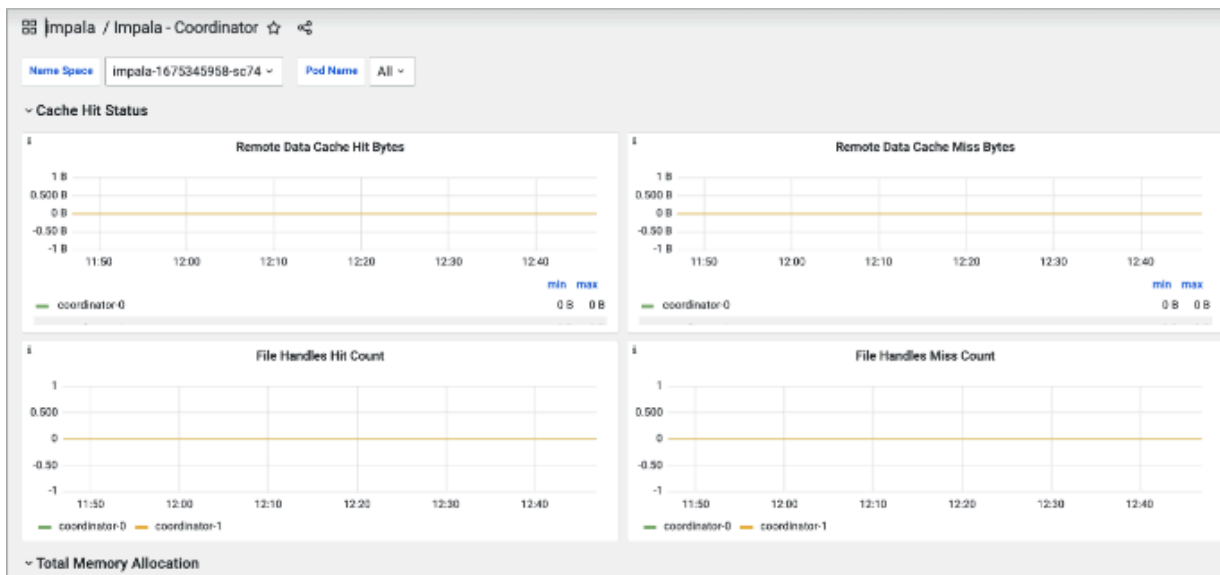
Monitoring Impala coordinators

You can monitor Impala coordinators caching, resource use, such as files handles hit, memory consumption, and more.

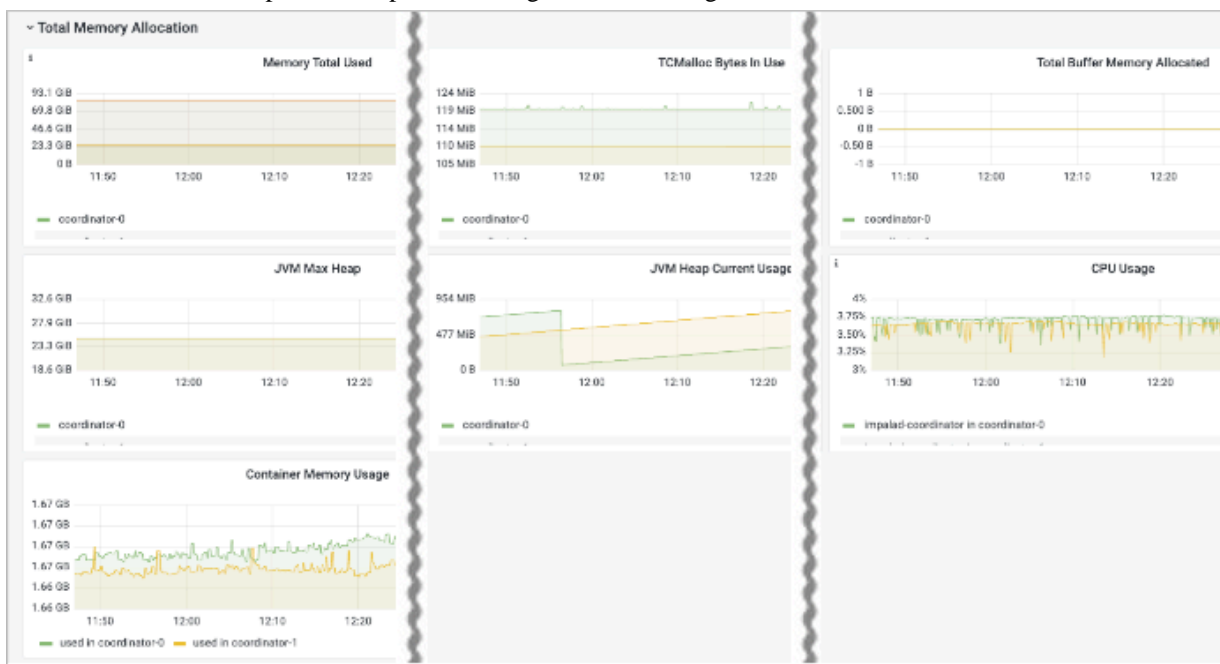
Procedure

1. In the list of Impala dashboards, click Impala - Coordinator.

2. Select a name space and pods, and view cache hit status, remote data cache miss bytes, catalog cache hit rate (not shown), file handles hit count, and file handles miss count.




3. Scroll down to Total Memory Allocation and view memory used, TCMalloc bytes in use, total buffer memory allocated, JVM max heap, JVM heap current usage, and CPU usage.

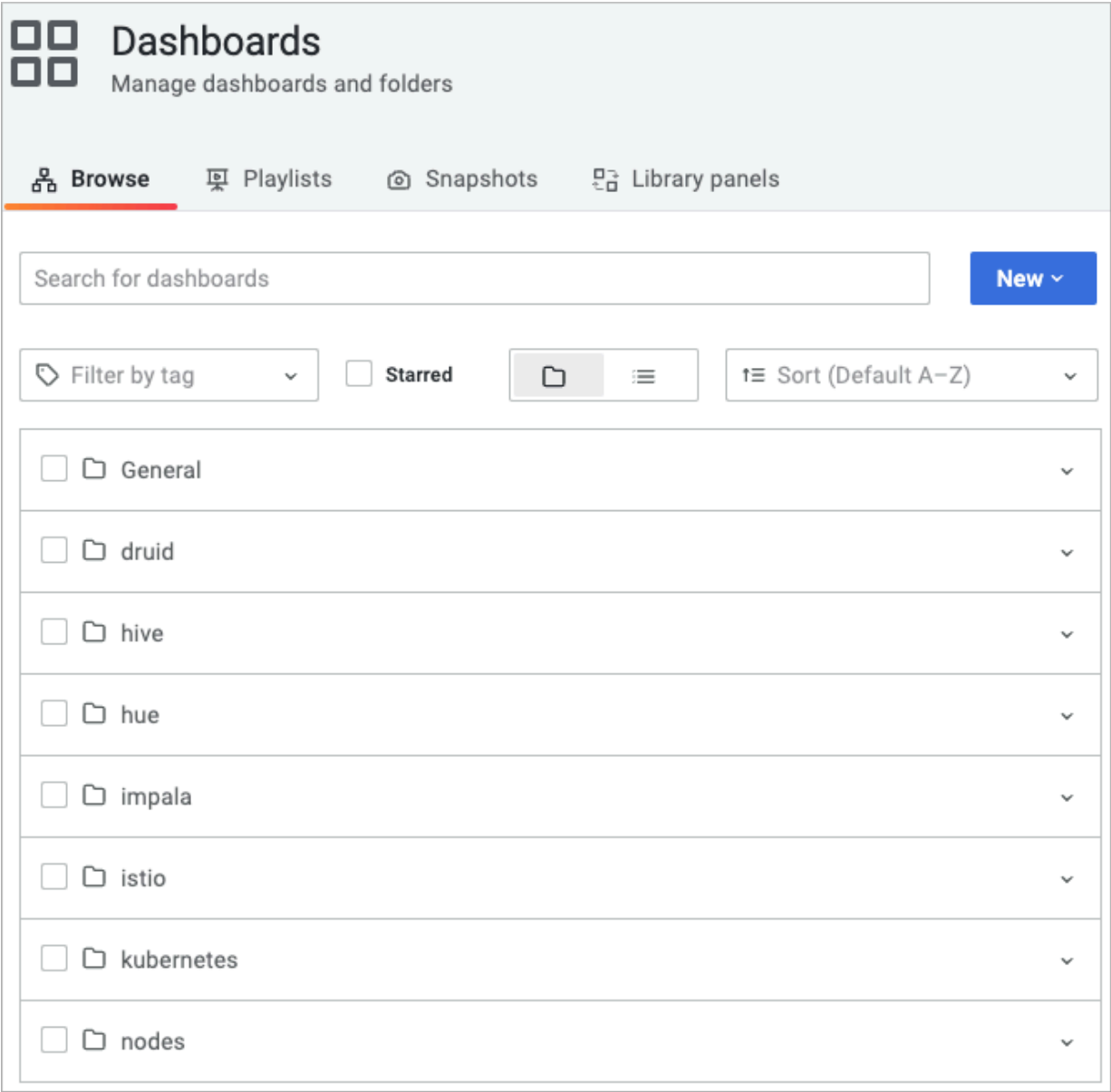


Monitoring nodes

You can get Kubernetes, cluster health, and node health information by monitoring Cloudera Data Warehouse nodes.

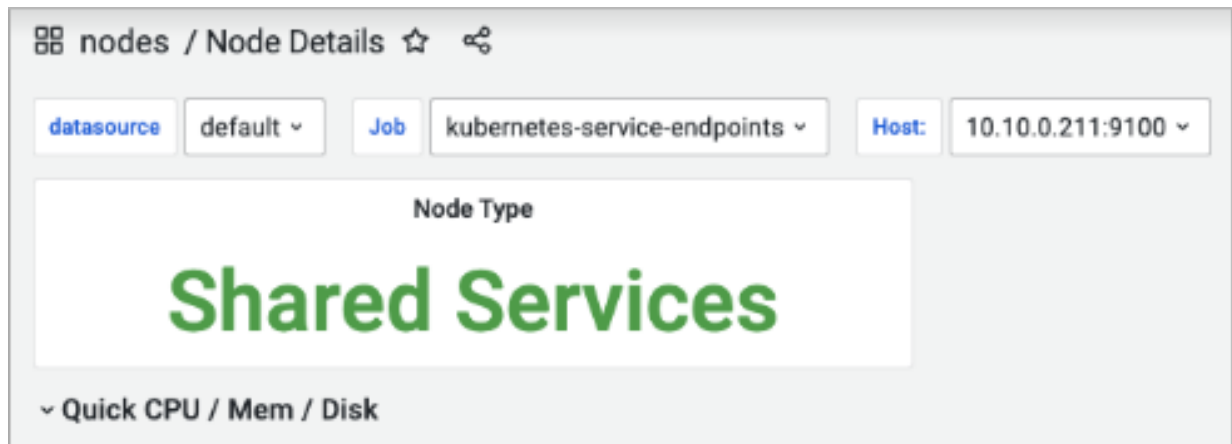
Procedure

1.
- In the Welcome screen, click grid , and then select Manage.
A list of dashboard groups appears:

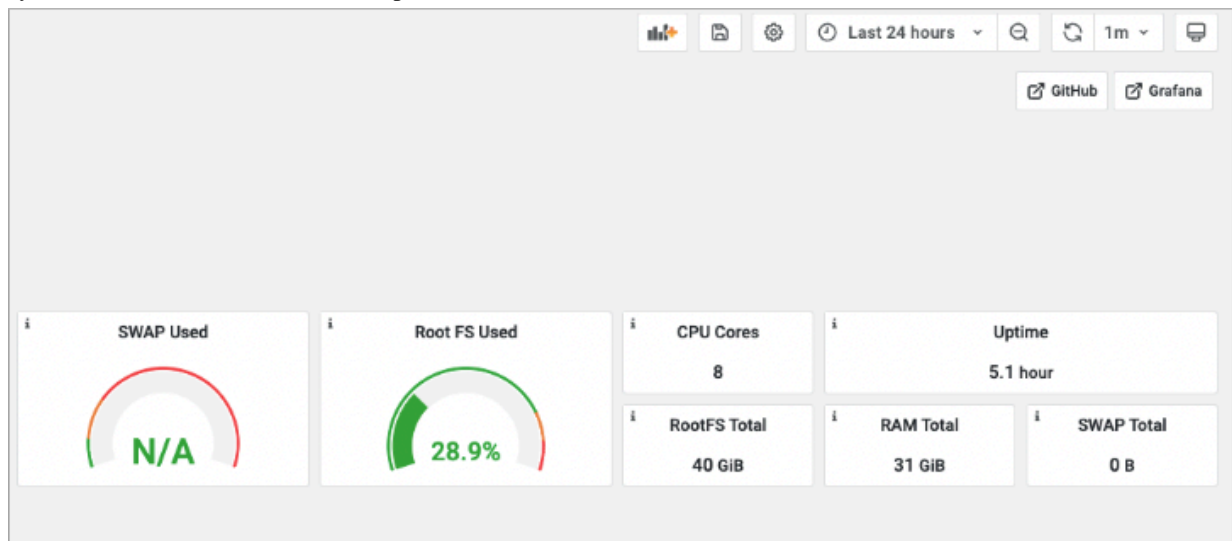


2. Click the nodes dashboard group.
Names of the nodes dashboards in the group appear:

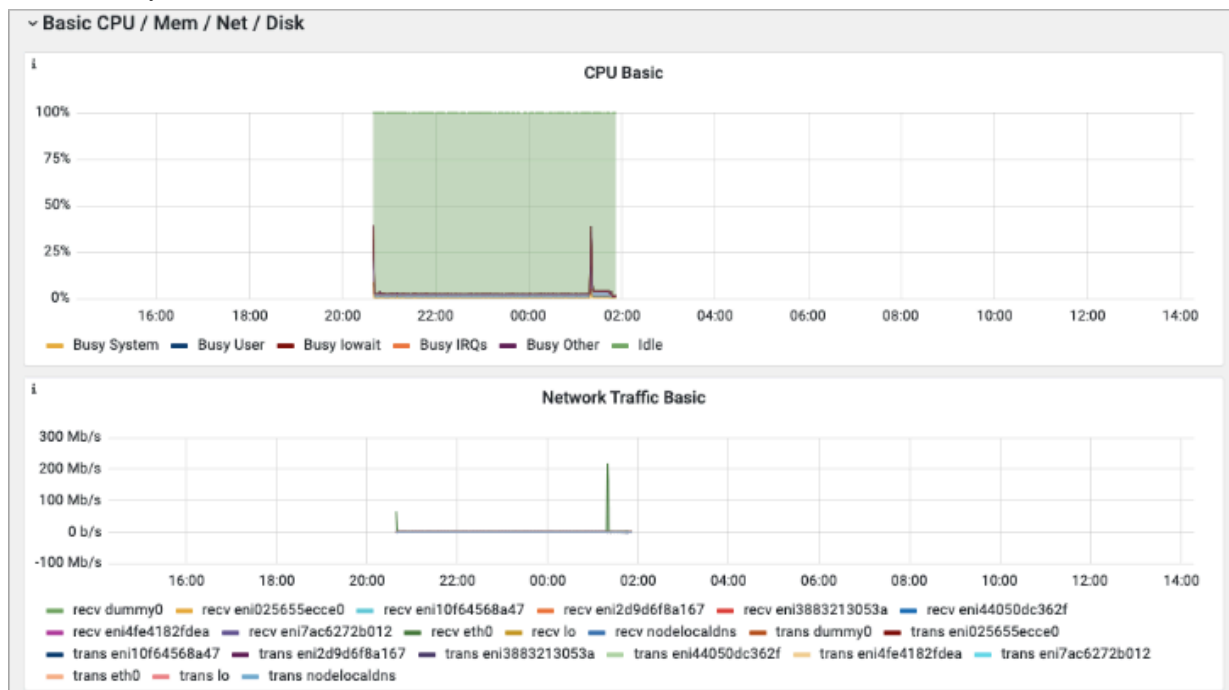
3. In the list of nodes dashboards, click Node Details, and then set the node of interest: In datasource, set default for example; in Job, set kubernetes-service-endpoints for example; in Host, set the IP address of a host in the Cloudera Data Warehouse cluster.



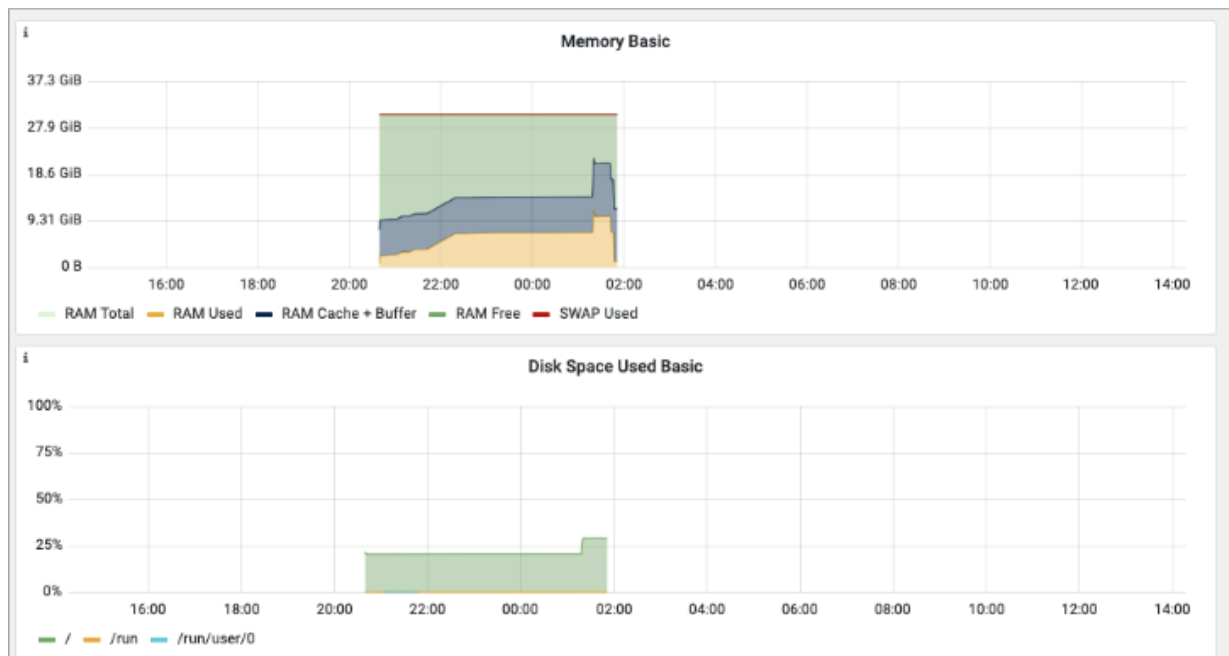
4. Expand Quick CPU/Mem/Disk, and view the percentage of busy CPU, the percentage of average load over the past 5 minutes and over the last 15 minutes, and RAM used.
5. On the right side of the dashboard, with Last 24 hours selected, for example, view the SWAP space used, root file system used, CPU cores used, and uptime of the node.



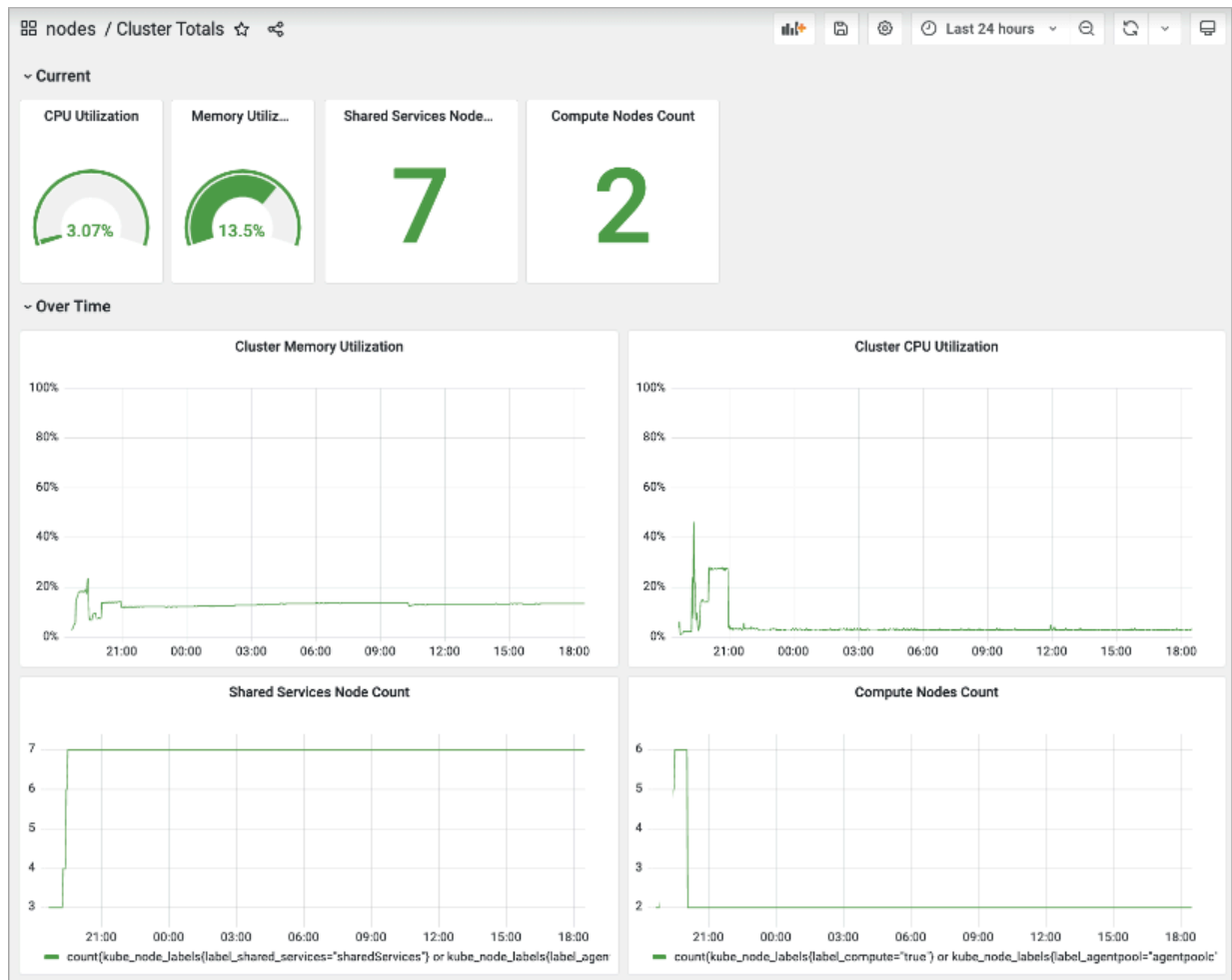
6. Scroll down, expand Basic CPU/Mem/Net/Disk, and see graphs on the left showing metrics about high activity from various sources, such as a busy user or busy interrupt requests (IRQ), and a graph of packets received and transmitted by network traffic.



7. On, the right, see graphs of basic memory and disk space use.



8. In the list of nodes dashboards, click Cluster Totals to view cluster health in graphs and metrics over the past 24 hours, as indicated by Cluster Memory Utilization, Cluster CPU Utilization, Shared Services Node Count, and Compute Nodes Count.



9. In the list of nodes dashboards, click Node Trends to view node health over the past 24 hours, as indicated by CPU Utilization, Load Average and CPU Count, Memory Utilization, and Disk Utilization.



Monitoring Kubernetes Services from Grafana

You can monitor the Amazon Elastic Kubernetes Service (EKS) or the Azure Kubernetes Service (AKS) used by your Cloudera Data Warehouse cluster from Grafana.

About this task

You can also monitor the Amazon Elastic Kubernetes Service (EKS) from the [Cloudera K8s dashboard](#).

Procedure

1.
- In the Welcome screen, click grid , and then select Manage.
A list of dashboard groups appears:

Dashboards

Manage dashboards and folders

Browse

Playlists

Snapshots

Library panels

Search for dashboards

New

Filter by tag

Starred

Sort (Default A-Z)

General

druid

hive

hue

impala

istio

kubernetes

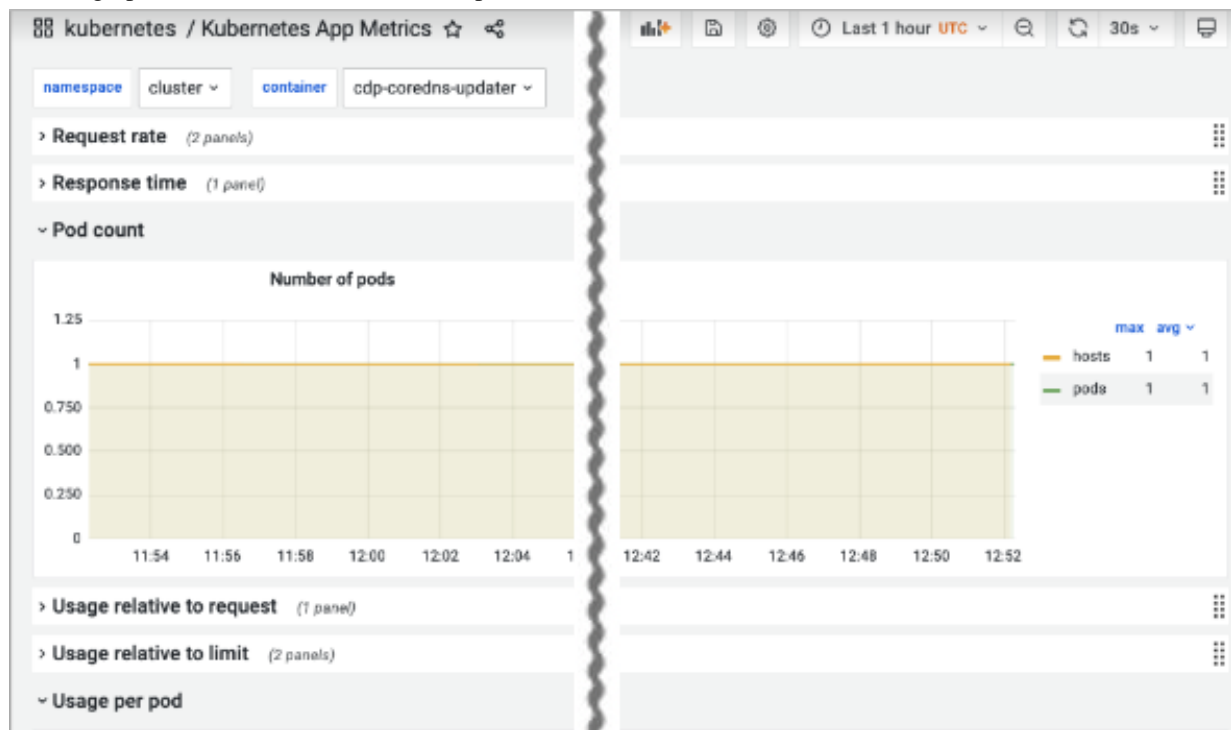
nodes

2. Click the kubernetes dashboard group.
Names of the kubernetes dashboards in the group appear:

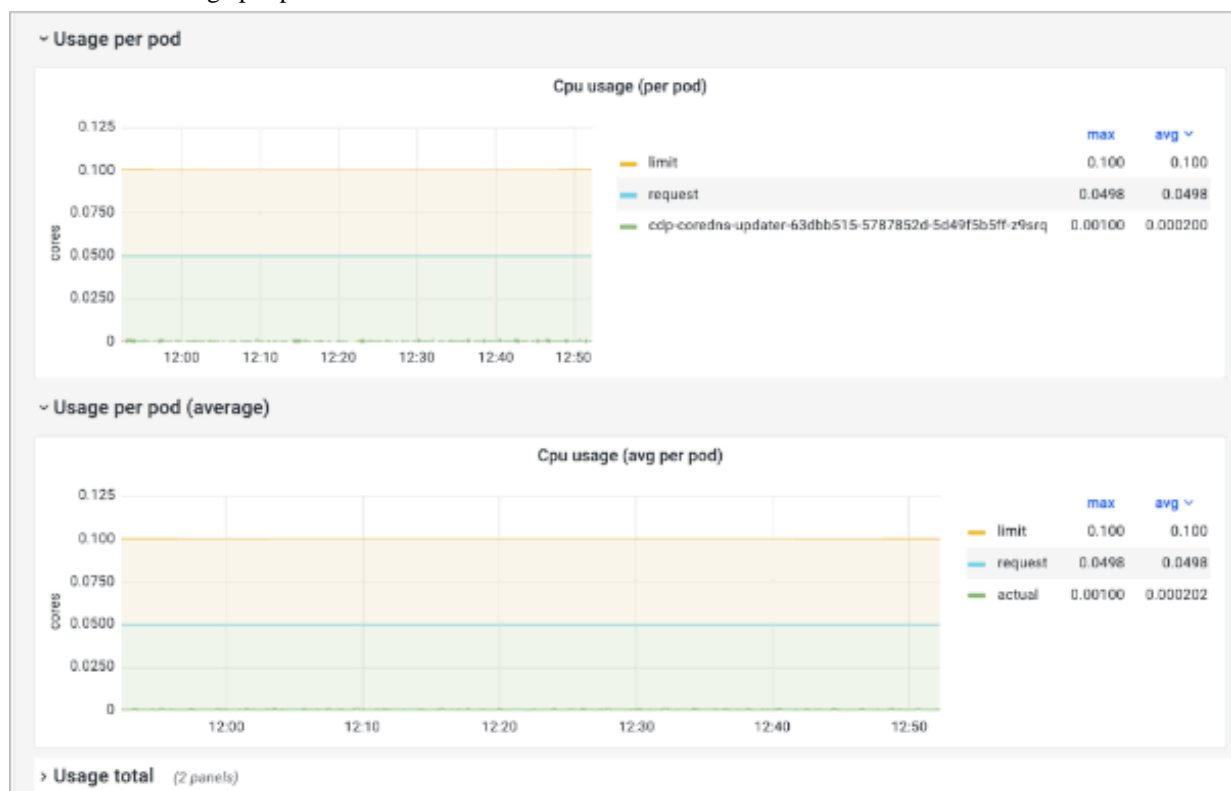
35

- Click the Kubernetes App Metrics, and then select a namespace, for example cluster, and container, for example a container named cdp-coreDNS-updater.

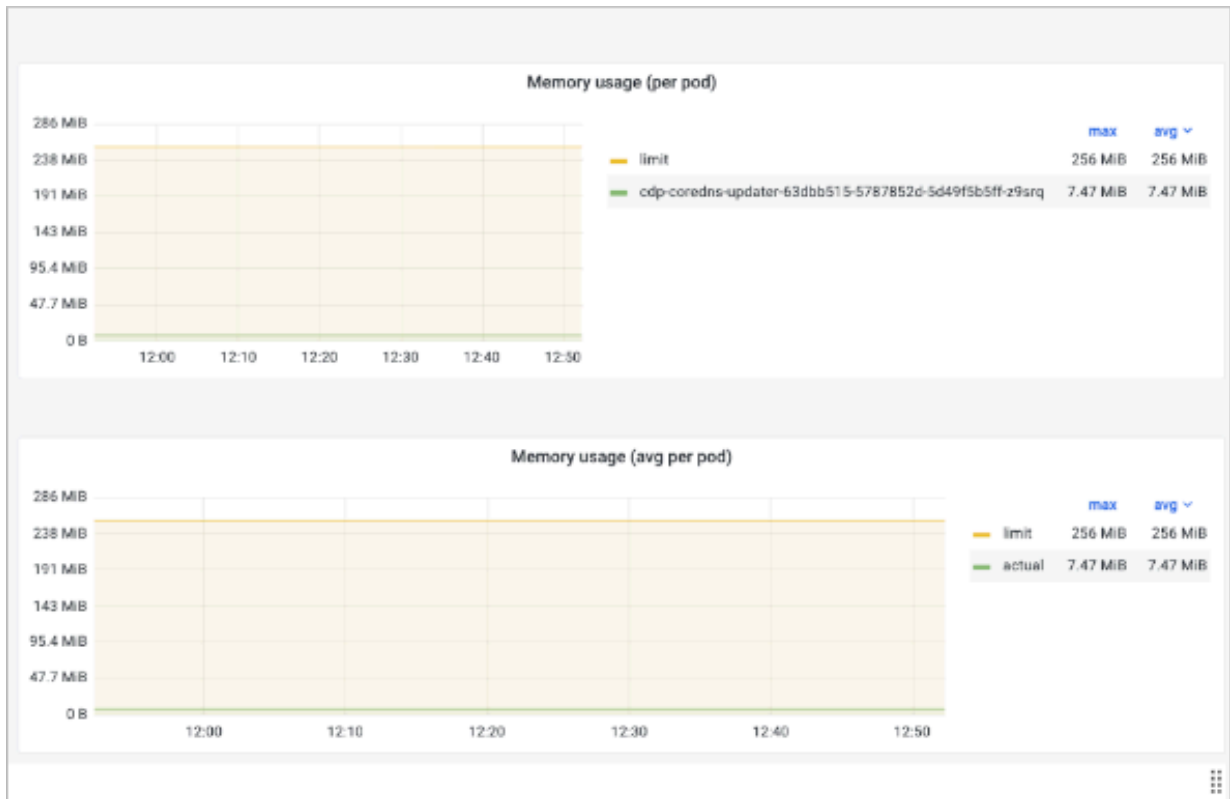
View a graph and metrics of the number of pods.



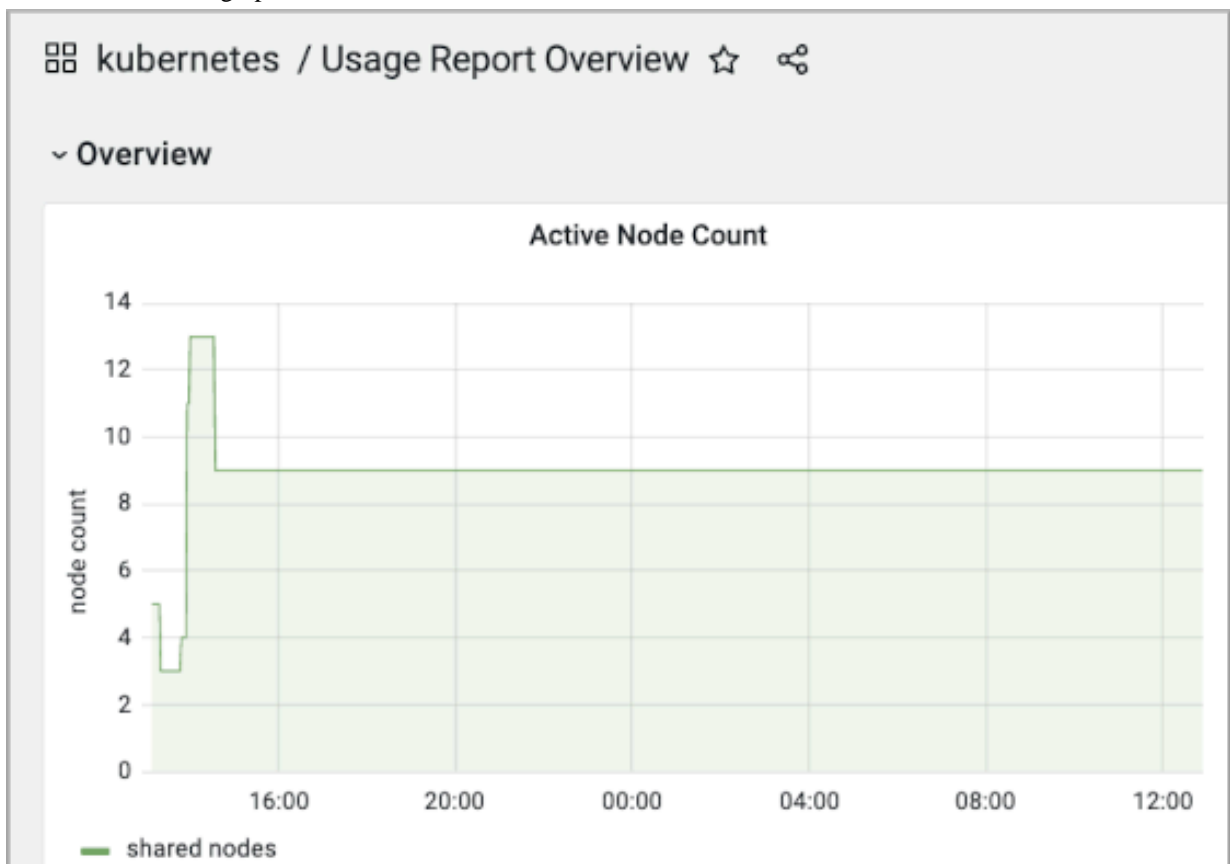
- View the CPU usage per pod.



5. On the right, view the memory usage per pod.



6. From the kubernetes dashboard group, click Usage Report Overview, and see the active nodes on the X axis of the Active Node Count graph.

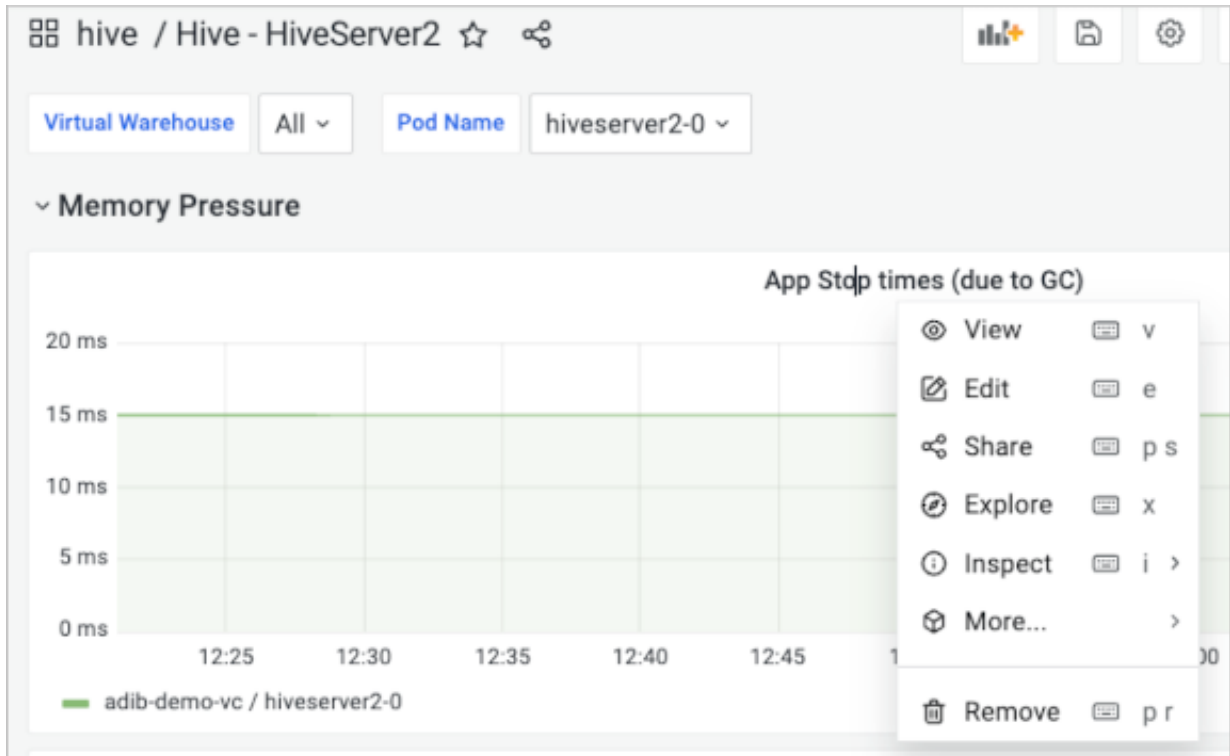


Updating a dashboard graph

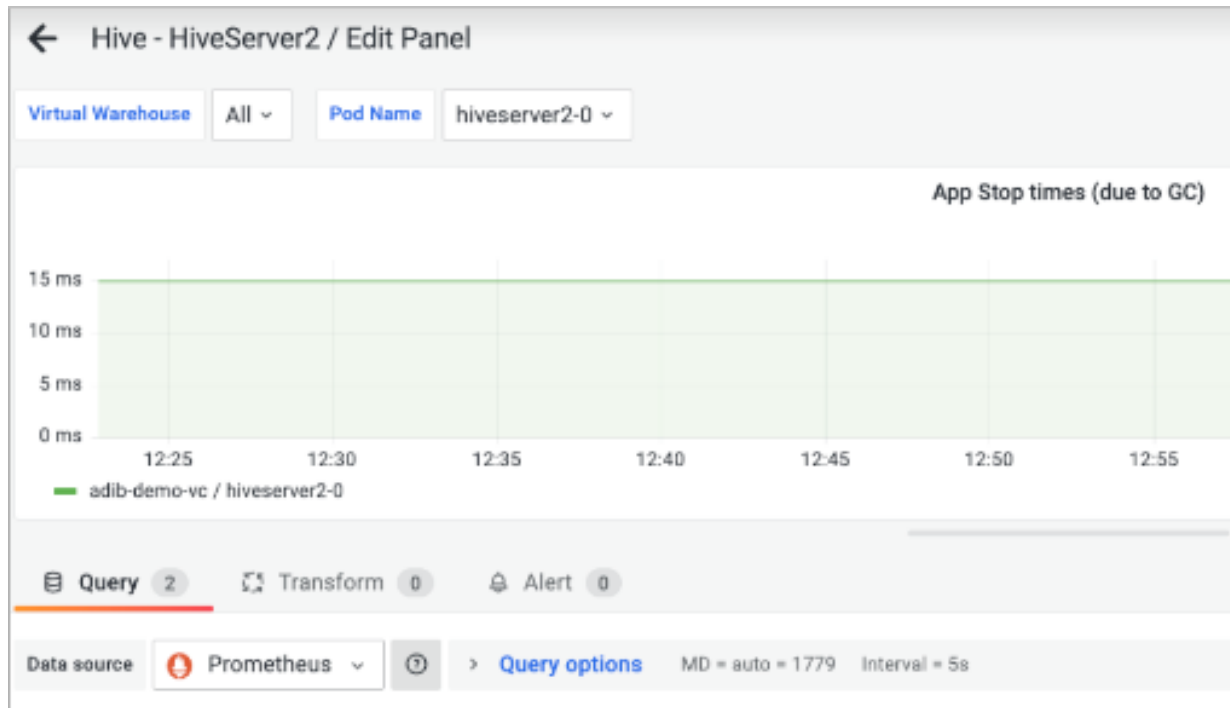
You can view, edit, share, explore, or remove a dashboard graph, and more.


Procedure

1. Open a dashboard, for example the HiveServer2 dashboard, and click the title in a row, for example, the **App Stop times** title.





2. Click Edit.
The edit panel appears.



3. At the top left side of the edit panel, change the Virtual Warehouse and Pod on which the graph is based.
4. At the top right side of the edit panel, click  to add variables and permissions to the panel.


5. On the right side of the edit panel, select a graph type, such as Time Series, and specify panel options, such as the title of the graph, tooltip mode, legend, graph styles, and more.

 Time series



Discard

Save

 Search options

All

Overrides

Panel options

Title

App Stop times (due to GC)

Description

Transparent background

> Panel links

> Repeat options

Tooltip

Tooltip mode

SingleAllHidden

Values sort order

NoneAscendingDescending

Legend

Legend mode

ListTableHidden

Legend placement

BottomRight

Legend values

Select values or calculations to show in legend

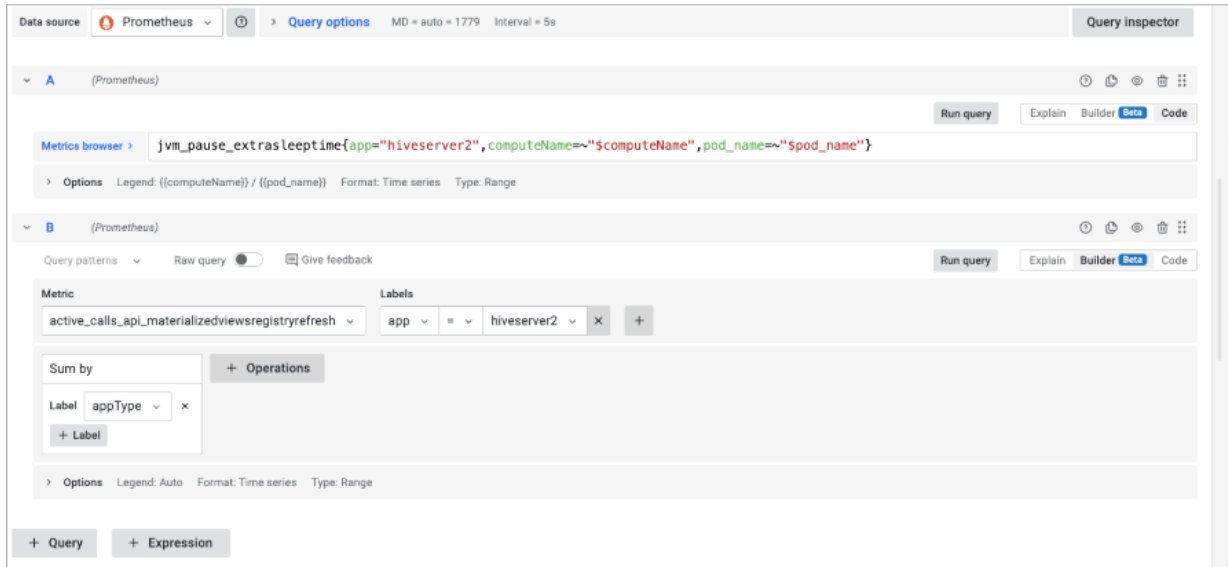
Choose

Graph styles

41

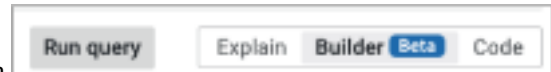
6. Scroll down the edit panel, and change query options.

For example, in Metrics browser, change the metrics to be graphed from `jvm_pause_extrasleeptime`.



- 7.

On the right side of the edit panel, use the Builder code option to frame the query and Explain to understand the query.



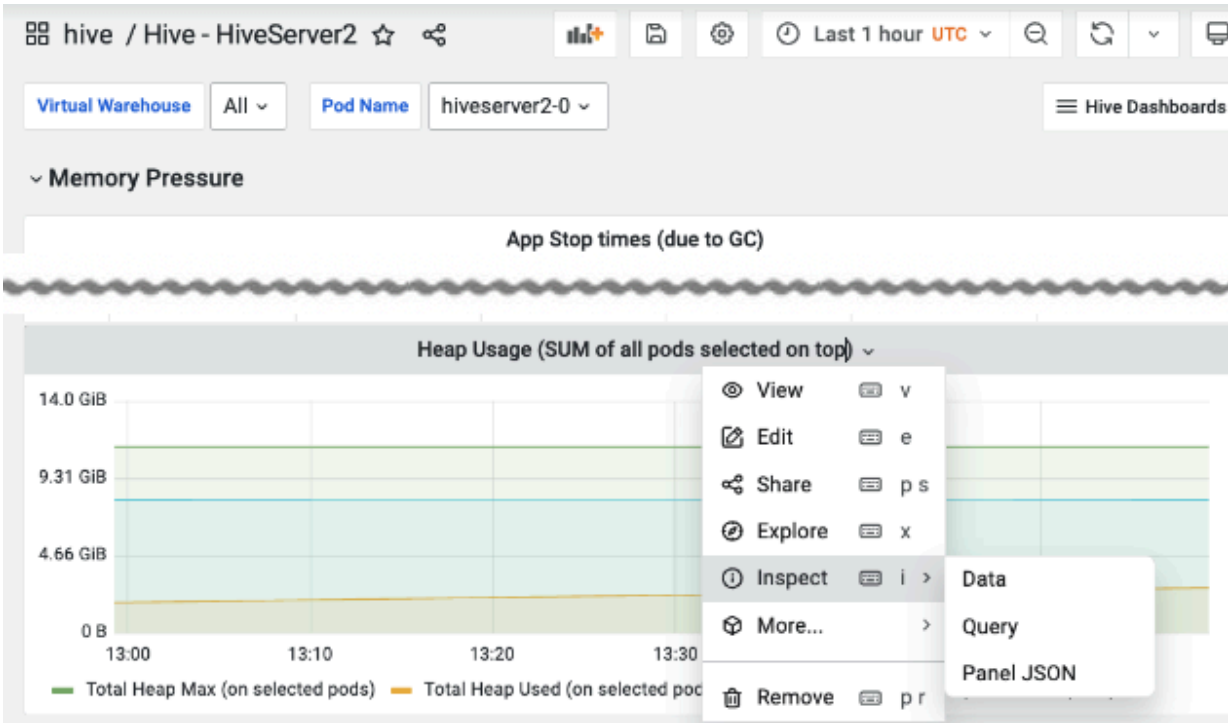
8. At the bottom of the edit panel, click + Query to add a query.
9. At the top-right, click Apply to save changes.

Inspecting dashboard data and queries

You can drill down into details of resource usage of Hive or Impala, get statistics about usage, and run queries on the details.

Procedure

- 1. Open a dashboard, for example the HiveServer2 dashboard, and click the title in a row, for example, the **Heap usage** title.



- 2. Click **Inspect Data** .
The total max heap usage over time appears.

Inspect: Heap Usage (SUM of all pods selected on top)

3 queries with total query time of 85 ms

Data

Stats

JSON

Query

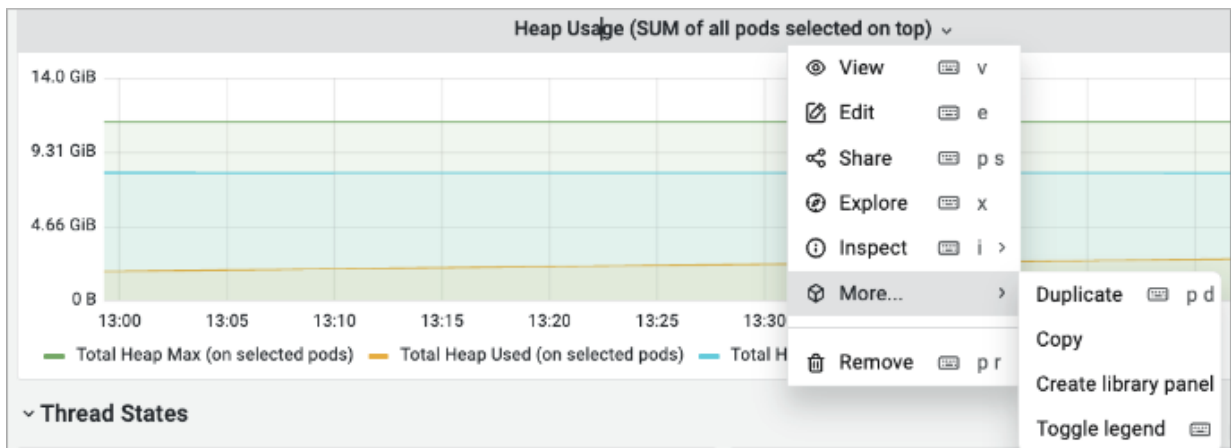
> Data op...

Total Heap Max (on selected pods), Formatted data

Download CSV

Time	Total Heap Max (on selected pods)
2023-02-03 12:59:15	12025069568
2023-02-03 12:59:20	12025069568
2023-02-03 12:59:25	12025069568

3. Click Stats or Query to view statistics or queries.
4. In the HiveServer2 dashboard, and click the Heap usage title More Duplicate panel content.



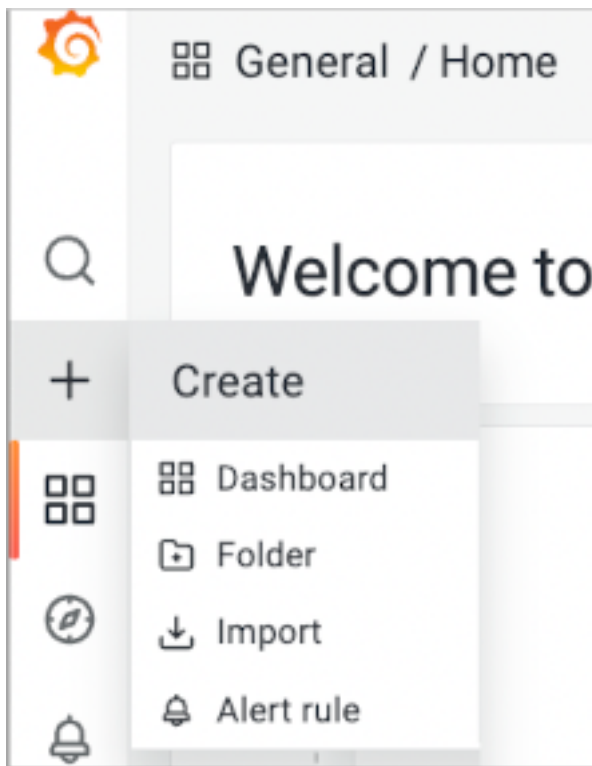
5. In the HiveServer2 dashboard, and click the Heap usage title More Copy panel content to another dashboard.

Creating a custom dashboard

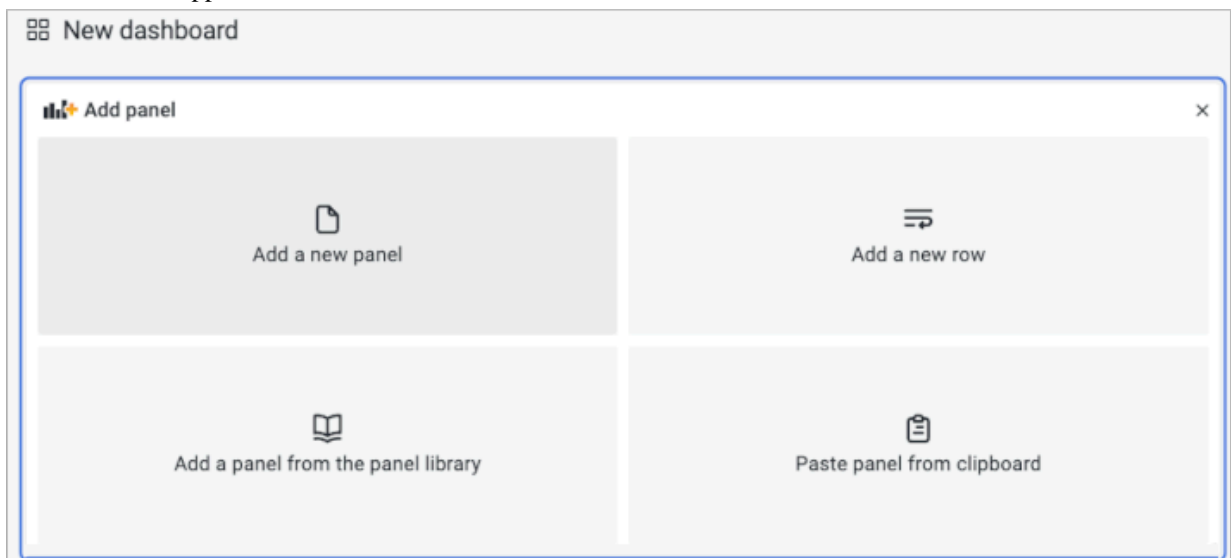
You can create a custom dashboard in a few steps.


Procedure

1. Log into Grafana, and in the Welcome screen, click + Create

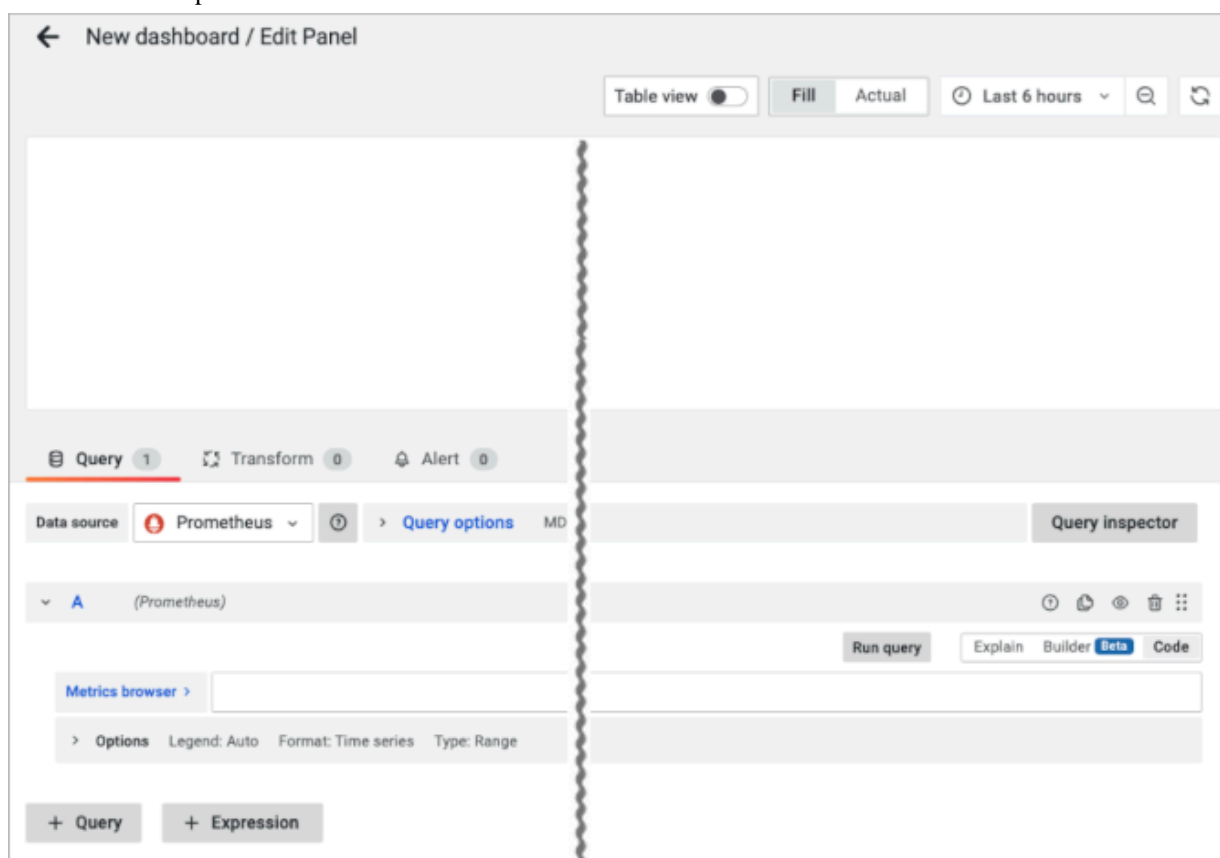


New dashboard appears.

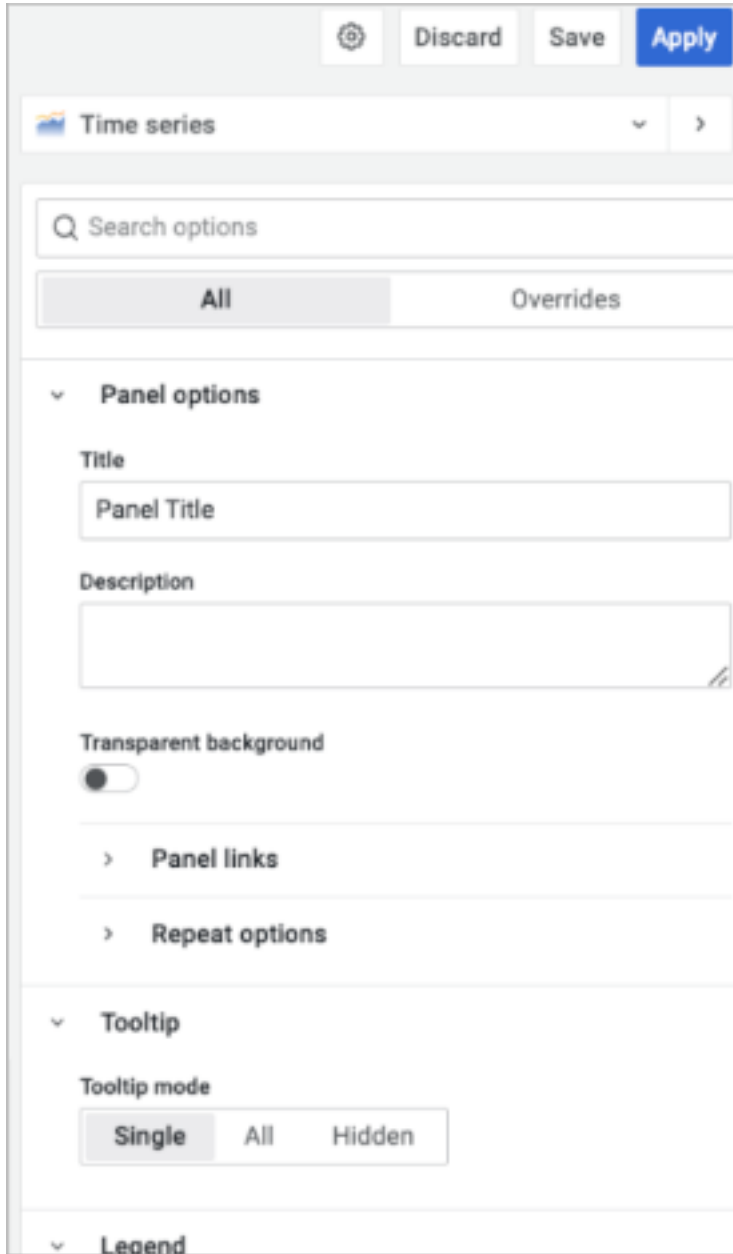


2. At the top right side of the edit panel, click  to add variables, permissions, and names to the new dashboard.

3. Click Add a new panel.



4. On the right side of the New edit panel, click  to add variables and permissions to the new dashboard.



The image shows the Grafana panel edit interface. At the top, there are buttons for 'Discard', 'Save', and 'Apply', along with a gear icon for settings. Below these is a 'Time series' dropdown menu. A search bar labeled 'Search options' is present. Two tabs, 'All' and 'Overrides', are visible. The 'Panel options' section is expanded, showing fields for 'Title' (with the value 'Panel Title') and 'Description'. A 'Transparent background' toggle switch is currently turned off. Below these are expandable sections for 'Panel links' and 'Repeat options'. The 'Tooltip' section is also expanded, showing 'Tooltip mode' with three options: 'Single' (selected), 'All', and 'Hidden'. At the bottom, the 'Legend' section is partially visible.

5. Edit the panel and rename it test1.

The screenshot shows the Grafana 'test / Edit Panel' interface. The main panel area displays a time series graph titled 'test1'. The graph shows a single green line representing the query `max(jvm_pause_extrasleeptime(app='metastore'))`. The x-axis shows time from 14:00 to 16:00. Below the graph, the 'Query' tab is selected, showing the data source as 'Prometheus' and the query text. The 'Options' section shows 'Legend: Verbose' and 'Format: Time series'. On the right, the 'Panel options' sidebar is open, showing the title 'test1' and a description field. Other options like 'Transparent background', 'Panel links', 'Repeat options', and 'Tooltip' are also visible. At the top right, there are buttons for 'Discard', 'Save', and 'Apply'.

6. Click Apply.
Test1 is added to the dashboard.

7. Edit the panel, rename it test2, and click Apply.
Test2 is added to the dashboard with Test1.



Meeting prerequisites to set up alerts

Before you can set up an alert triggered by a dashboard event, you must configure an SMTP (Simple Mail Transfer Protocol) server, an alert recipient, and a notification policy.

Procedure

1. Configure grafana using kubectl.

```
kubectl edit configmap grafana -n istio-system
```

For more information, see ["Granting remote access to Kubernetes"](#).

2. Update the grafana.ini to configure SMTP, setting user to the alert account email id and the password to the app password generated in your gmail account.

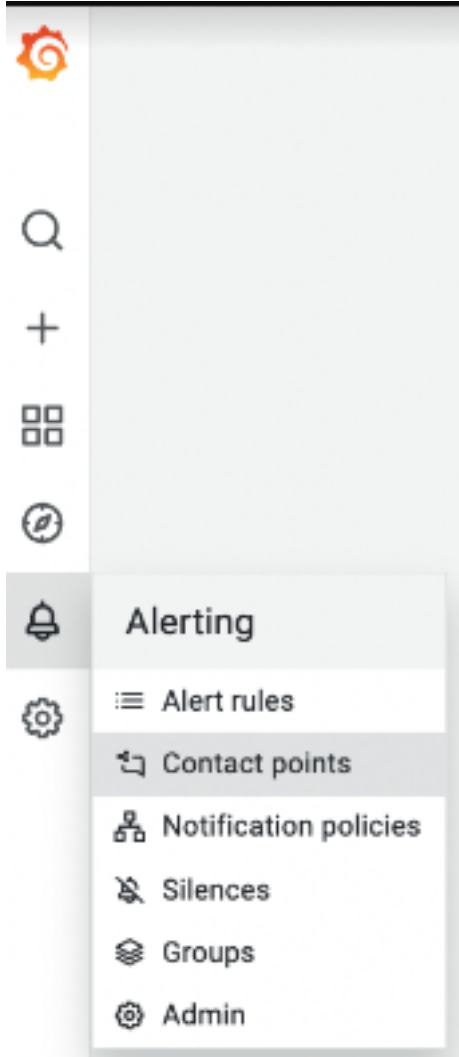
```
...
[smtp]
enabled = true
host = smtp.gmail.com:465
user = youremail@gmail.com
password = substitute_your_password
;cert_file=
;key_file=
skip_verify = true
from_address = alertgrafan@gmail.com
from_name = Grafana
* EHLO identity in SMTP dialog (defaults to instance_name)
;ehlo_identity = dashboard.example.com
```

```
;startTLS_policy = NoStartTLS
```

3. Refresh the grafana pod.


```
kubectl get pod -A grep -i grafana  
kubectl delete pod grafana-9bcb5d5c-w8r6c -n istio-system
```

4. Log into Grafana, and click Alerting Contact points .



5. In Contact point type, select an alert receiver type.

6. In Addresses, enter an email address, and then in Message, compose an email message body. For example, enter an email message body test grafana email.



Alerting

Alert rules and notifications

Alert rules

Contact points

Notification policies

Silences

Alertmanager

Grafana

Update contact point

Name *

grafana-default-email

Contact point type

Email

Test

Addresses

You can enter multiple email addresses using a "," separator

amit.mishra@cloudera.com

Optional Email settings

☐ Single email

Send a single email to all recipients

Message

Optional message to include with the email. You can use template variables

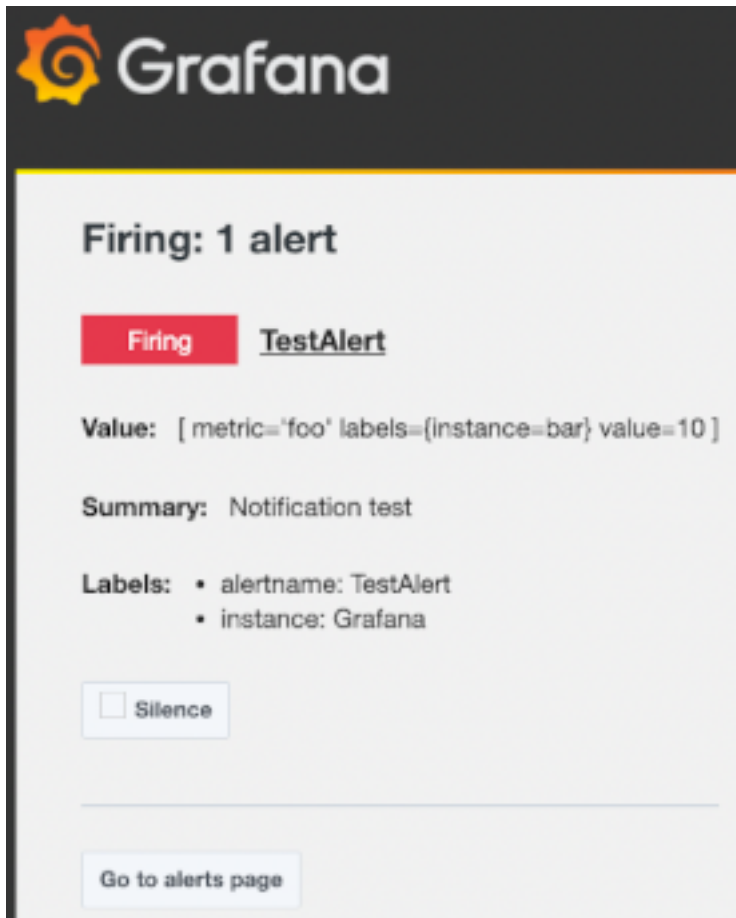
test grafana email

Notification settings

☐ Disable resolved message

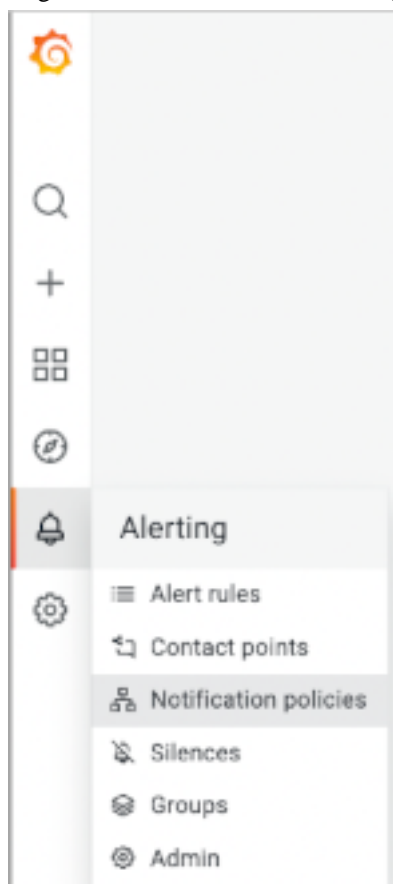
Disable the resolve message [OK] that is sent when alerting state returns to false

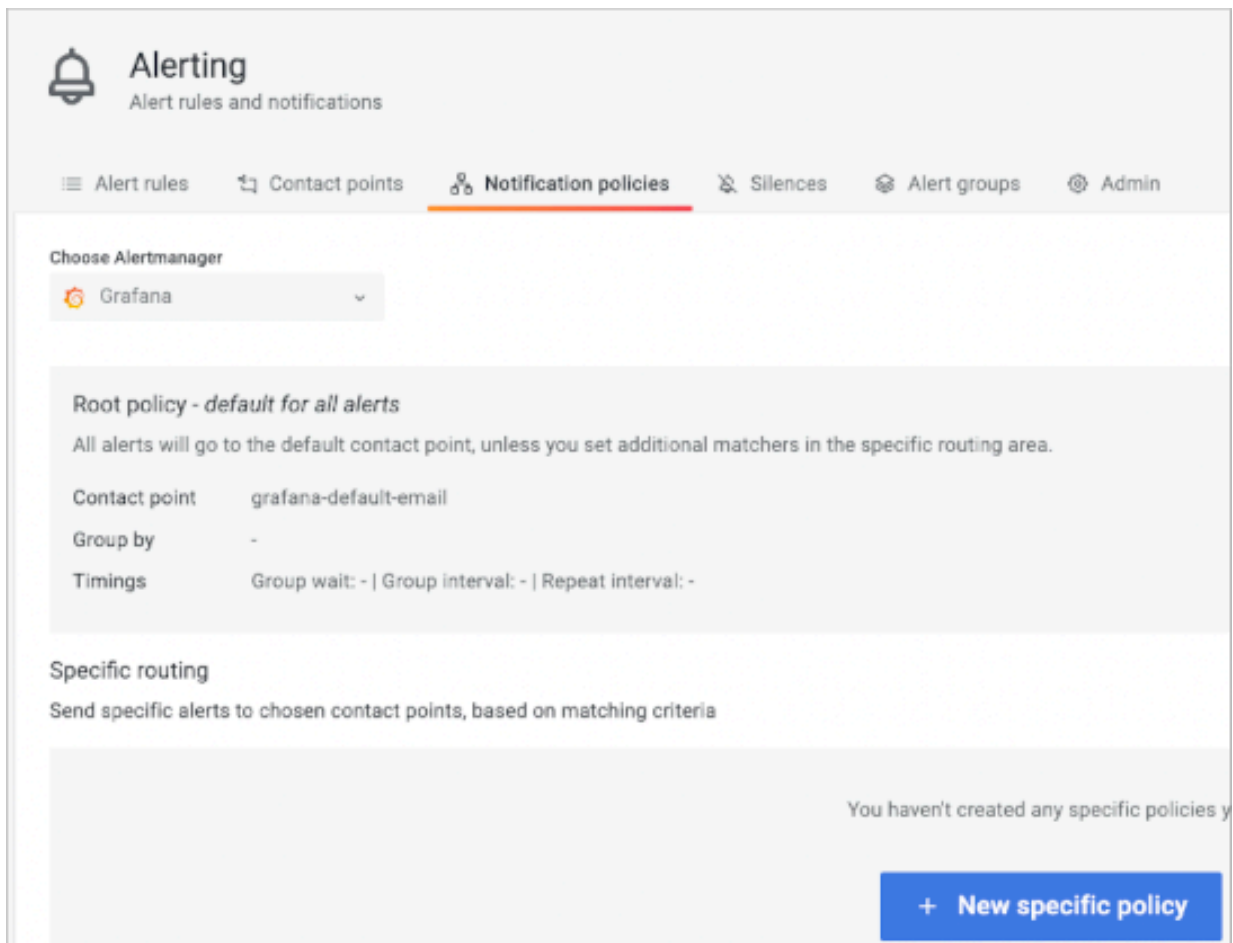
- Click Test to verify that a test alert arrives in your inbox.



- Scroll up and in Name, enter the contact point name.

9. Log into Grafana, and click Alerting Notification policies .




10. Click New specific policy.

The screenshot shows the Grafana Alerting interface. At the top, there's a header with a bell icon and the word 'Alerting' followed by 'Alert rules and notifications'. Below this is a navigation bar with tabs: 'Alert rules', 'Contact points', 'Notification policies' (which is selected and highlighted with an orange underline), 'Silences', 'Alert groups', and 'Admin'. Under the 'Notification policies' tab, there's a section 'Choose Alertmanager' with a dropdown menu showing 'Grafana'. Below this, there's a section titled 'Root policy - default for all alerts' with a description: 'All alerts will go to the default contact point, unless you set additional matchers in the specific routing area.' This section contains a table with the following details:

Contact point	grafana-default-email
Group by	-
Timings	Group wait: - Group interval: - Repeat interval: -

Below the root policy section is a 'Specific routing' section with the text 'Send specific alerts to chosen contact points, based on matching criteria'. This section is currently empty, displaying the message 'You haven't created any specific policies yet'. At the bottom right of this section is a blue button with a plus sign and the text '+ New specific policy'.

11. In Contact Point, select the contact point to receive notification, and save the policy.



Alerting


Alert rules and notifications

Alert rules

Contact points

Notification policies

Choose Alertmanager

 Grafana

Root policy - default for all alerts

All alerts will go to the default contact point, unless you set additional

Contact point

grafana-default-email

Group by

-

Timings

Group wait: - | Group interval: - | Repeat interval: -

Specific routing

Send specific alerts to chosen contact points, based on matching criteria

Matching labels

Matches all alert instances

Matching labels

If no matchers are specified, this notification policy will handle all alert

+ Add matcher

Contact point

grafana-default-email

Continue matching subsequent sibling nodes

☐

Override grouping

☐

Override general timings

☐

Mute timings

Add mute timing to policy

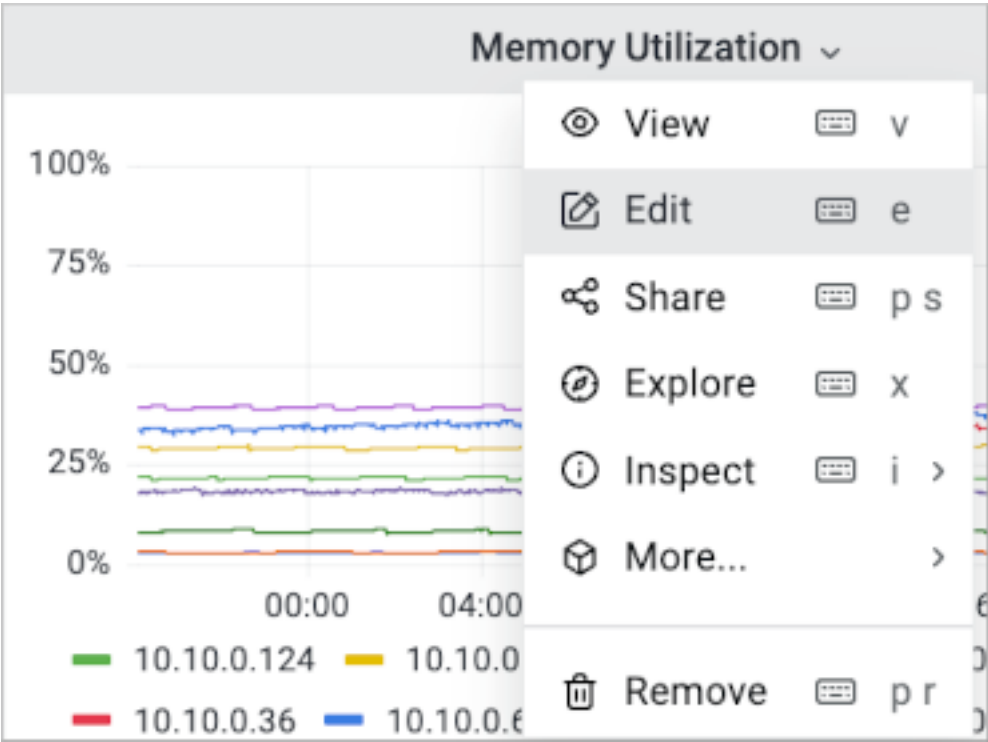
Choose

Creating alerts

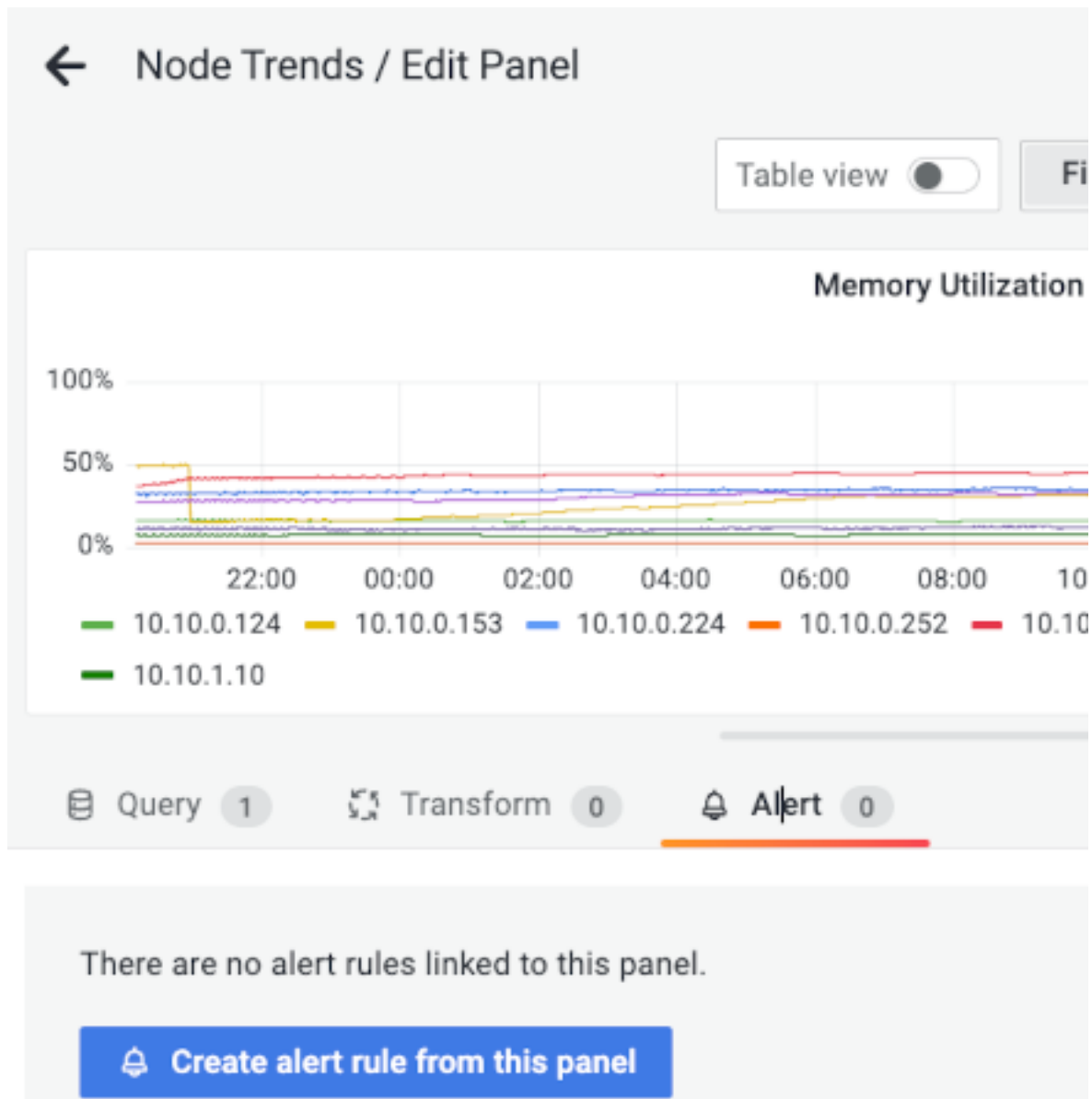
After meeting prerequisites to set up alerts, you can create an alert for an existing dashboard.

Procedure

1. Click the title of the graph that you want to fire an alert, and select Edit.



2. In the edit panel that appears, click **Create alert rule** from this panel .



Create alert rule appears. The Rule name is set to Memory Utilization.

The screenshot shows the 'Create alert rule' dialog. It has a title bar with a bell icon and the text 'Create alert rule'. Below the title bar, there's a section '1 Rule type' with two options: 'Grafana managed alert' (selected) and 'Mimir or Loki alert'. The 'Grafana managed alert' option is highlighted with a blue border. Below the options, there's a text input field for 'Rule name' with the value 'Memory Utilization'. There's also a 'Folder' dropdown menu with the value 'nodes'. At the bottom right, there's a 'Group' section with a text input field containing 'test alert'.

3. In **Create alert rule** scroll down, and in Group, enter a meaningful name for the alert group, for example, test alert..

Rule name

Memory Utilization

Folder ⓘ
Select a folder to store your rule.

nodes

Group
Rules within the same group are evaluated after the same time interval.

test alert

2 Create a query to be alerted on

4. In **B**, define a classic condition to trigger the alert, for example when last() of A is above 20.

▼ B (_expr_)

Operation Classic condition

Conditions WHEN last() OF A IS ABOVE 20

+

5. In **Define alert conditions**, select B for the expression that triggers the alert.

3 Define alert conditions

Condition
The query or expression that will be alerted on

B

Evaluate
Evaluation interval applies to every rule within a group. It can overwrite the interval of an existing alert rule.

Evaluate every ⓘ 1m for ⓘ 5m

> Configure no data and error handling

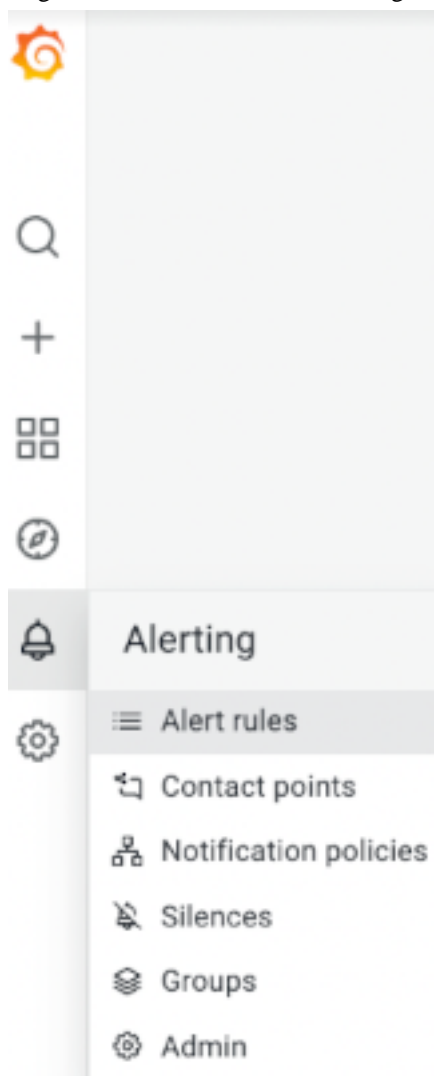
Preview alerts

6. Click Preview alerts, and then scroll up to the top right of **Create alert rule**, and click Save.


Reviewing alerts and notifications

Procedure

1. Log into Grafana, and click **Alerting** **Alert rules** .



- 2. In Alert rules, in the **hive** directory, click Firing to review the alert in the UI.



Alerting

Alert rules and notifications

Alert rules

Contact points

Notification policies

Search by data source

All data sources

Search by label ⓘ

Q Search

State

Firing

24 rules: 2 firing, 22 normal

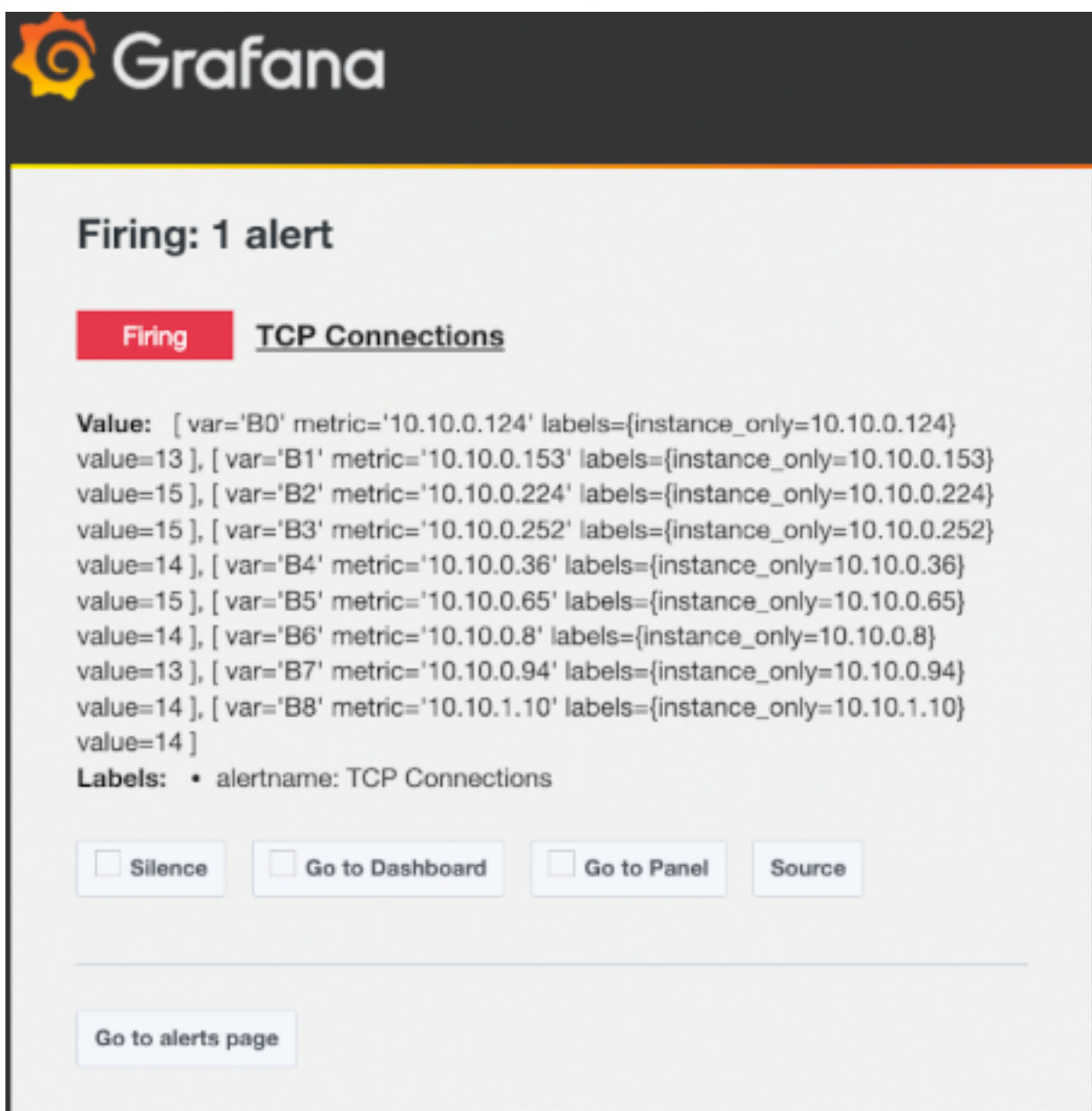
Grafana

hive

State	Name
> Firing for 24m	HiveServer2 Memory - Max

nodes

State	Name
> Firing for 26m	TCP Connections



Forwarding Prometheus metrics from Cloudera Data Warehouse to an endpoint

You can configure Prometheus in Cloudera Data Warehouse to push its metrics to an external endpoint, such as Prometheus, Grafana, Thanos, or some other endpoint.

About this task


For information about other receivers, see Prometheus documentation: [Compatible senders and receivers](#).

In this procedure you add a configuration snippet that triggers a Prometheus Remote Write. Prometheus then sends HTTP POST requests to the Remote Write endpoint to forward your metrics.

Before you begin

Cloudera Data Warehouse must run in an AWS or Azure environment.

Procedure

1. In **Environments**, locate your Cloudera Data Warehouse environment.
2. Click , and select **Edit Observability**.
3. In **Metrics Forwarding Configuration**, add the config and authentication methods for your endpoint in YAML format.
For example:

```
remote_write:
- url: https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-3e8fcedb-6727-4912-82aa-f05a8be25fe4/api/v1/remote_write
  sigv4:
    region: us-west-2
    name: prometheus_remote_write
  queue_config:
    max_samples_per_send: 1000
    max_shards: 200
    capacity: 2500
```

4. Click **Apply Changes**.

Results

This action updates the config map. The Prometheus pod restarts, so the new config takes effect.

Monitoring Kubernetes resources from K8s dashboard

As a Cloudera Data Warehouse administrator, you can monitor Kubernetes resources in your Cloudera Data Warehouse cluster. On the K8S dashboard, you can view the state of the resources, such as CPU and memory usage, see the status of pods, and download logs.

The dashboard can provide insights into the performance and health of a Cloudera Data Warehouse cluster. From the dashboard, if authorized, you can monitor the environments of any Cloudera Data Warehouse cluster efficiently. You do not need to copy/paste kubeconfig files to switch to monitoring another environment. Monitoring the dashboard can help keep your cluster running smoothly and efficiently.

By default, the dashboard is disabled. You follow instructions in the next topic to activate and use the dashboard. Using the dashboard incurs some cloud cost, and is designed to time out automatically after 4 hours to prevent wasting resources. Cloudera recommends deactivating the dashboard when not in use to reduce cloud expenses.

Limitations

Cloudera Data Warehouse does not support using the Kubernetes dashboard in environments with internal load balancer (Enable internal load balancer (ingress) option) on Azure, and with Private Load Balancer mode in AWS.

Prerequisites

- You have an AWS or Azure environment in Cloudera Data Warehouse on cloud.
- You have [activated](#) your environment.
- You obtained the Cloudera Data Warehouse Admin role.


Activating the K8S dashboard

From your environment, you can flip a switch to activate, or deactivate the K8S dashboard. You can then click a link to view the K8S dashboard in your web browser.

Before you begin

You must meet the prerequisites listed in the previous topic.

Procedure

1. In the Cloudera Data Warehouse service, go to the Environments tab.
2. Locate the environment that you want to view.
3. Click  Edit .
The **Environment Details** page is displayed.
4. Toggle Activate Dashboard to on.
5. Wait for the K8S Dashboard Activated success message to appear.
A link to the K8S dashboard appears in your environment tile.

What to do next

- Follow steps in the next topic to familiarize yourself with the K8S dashboard.
- Deactivate the dashboard when not in use because an activated dashboard incurs a cost.


Using the K8S dashboard

You see how to view the dashboard and get ideas about the types of K8S metrics, charts, and other visualizations that appear on the dashboard. You might use the dashboard to keep your cluster running efficiently and troubleshoot problems.

Before you begin

- You have met the prerequisites listed in Monitoring Kubernetes resources.
- You have activated the K8S dashboard.

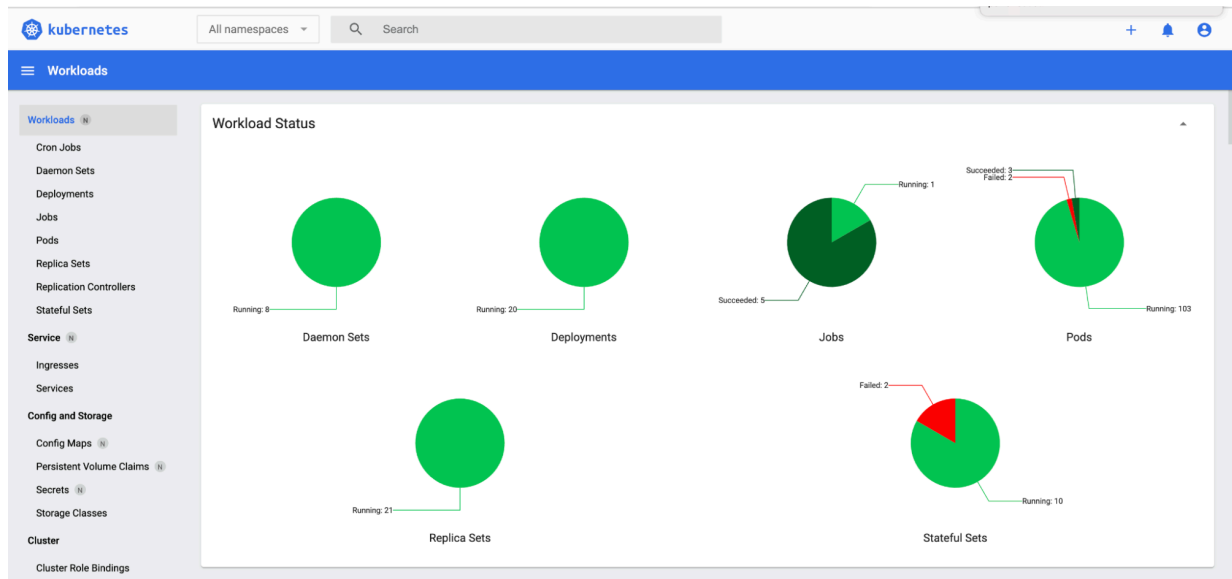
Procedure

1. Log in to the Cloudera web interface, navigate to Data Warehouse Overview , and select your environment.
Environment Details shows the K8S dashboard is activated.
2. In Environment Details, click Open .

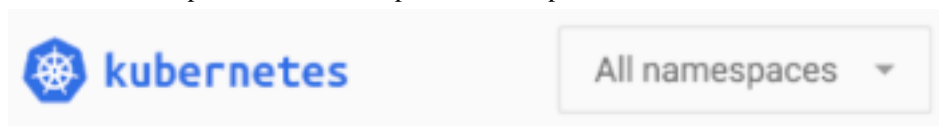


The K8S dashboard appears.

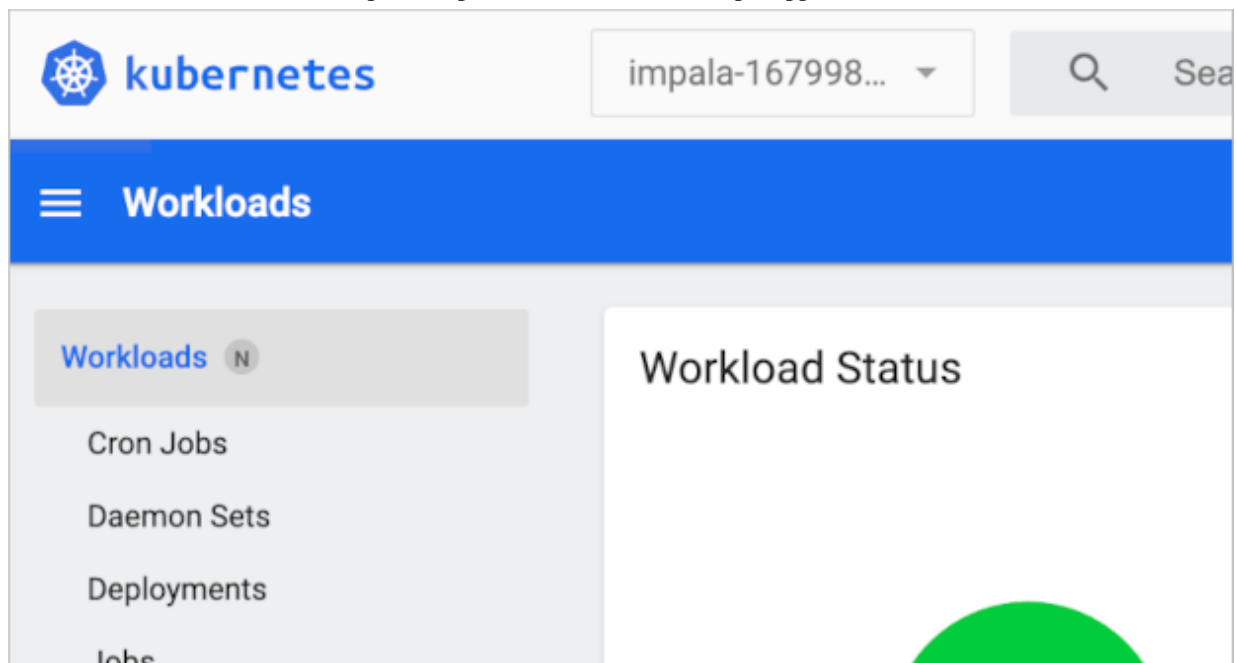
3. View the status of your containerized applications.
For example:



4. Click All Namespaces and select a specific Namespace.

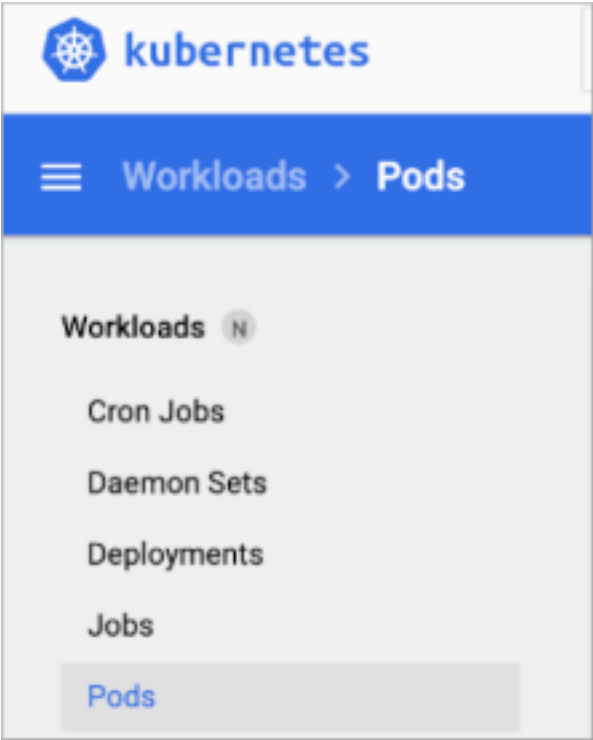


The workload status of the namespace, `impala-167998`, in this example appears.




5. Kubernetes dashboard features are available and documented on the [kubernetes site](https://kubernetes.io).

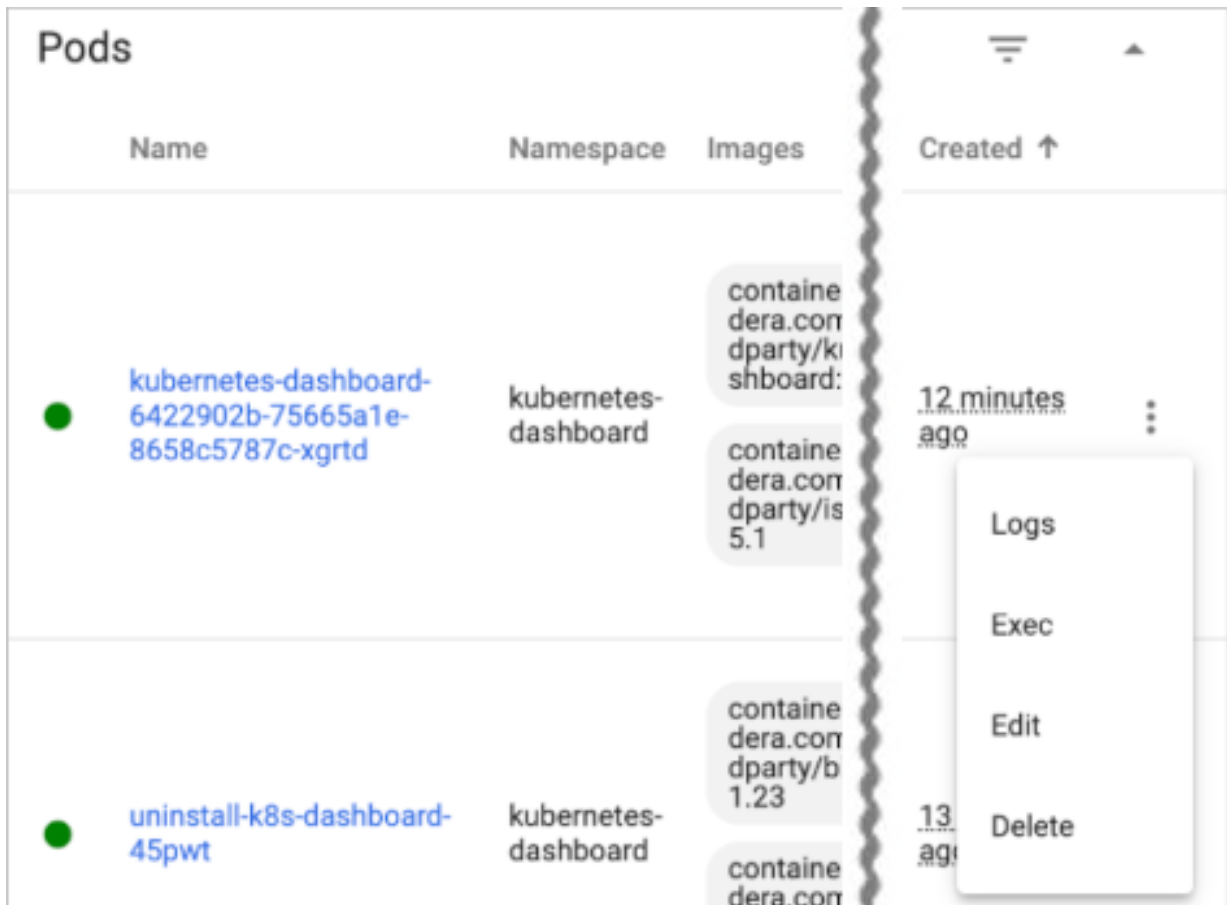
6. To download logs, click Pods.



The dashboard of pods appears.

Pods				
Name	Namespace	Images	Labels	No
<div><div></div><div>kubernetes-dashboard-6422902b-75665a1e-8658c5787c-xgrtd</div></div>	kubernetes-dashboard	<div><div>container-dev.repo.cloudera.com/cloudera_thirdparty/kubernetesui/dashboard:v2.5.1</div><div>container-dev.repo.cloudera.com/cloudera_thirdparty/istio/proxyv2:1.15.1</div></div>	<div><div>app.kubernetes.io/component: kubernetes-dashboard</div><div>app.kubernetes.io/instance: kubernetes-dashboard-6422902b-75665a1e</div><div>app.kubernetes.io/managed-by: Helm</div><div>Show all</div></div>	ip-172.17.0.2.c

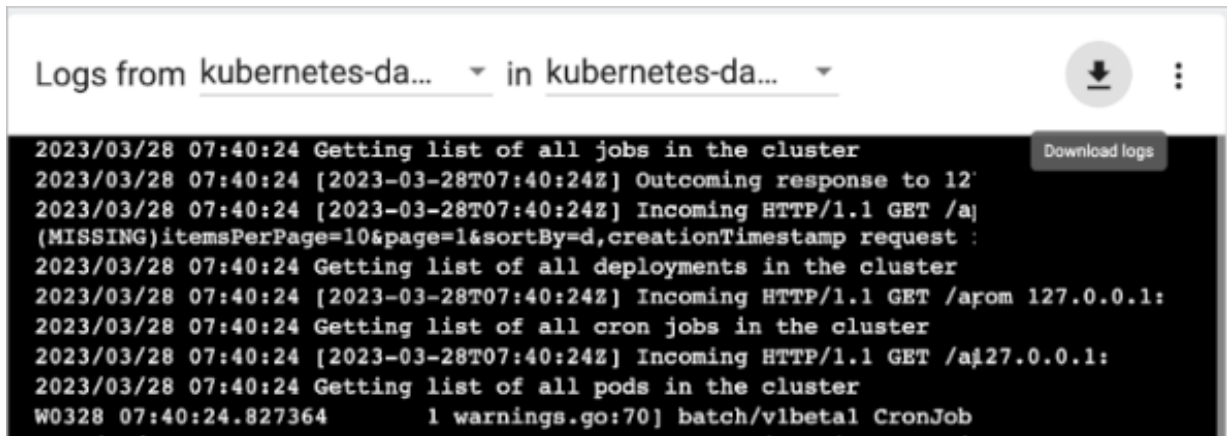
7. Click , and select Logs.
For example:



The screenshot shows the 'Pods' section of the Kubernetes dashboard. It lists two pods in the 'kubernetes-dashboard' namespace. The first pod, 'kubernetes-dashboard-6422902b-75665a1e-8658c5787c-xgrtd', is highlighted. A menu is open for this pod, showing options: Logs, Exec, Edit, and Delete. The 'Logs' option is selected.

Name	Namespace	Images	Created ↑
kubernetes-dashboard-6422902b-75665a1e-8658c5787c-xgrtd	kubernetes-dashboard	containere... dera.com dparty/k shboard:	12 minutes ago
uninstall-k8s-dashboard-45pwt	kubernetes-dashboard	containere... dera.com dparty/b 1.23 containere... dera.com	13 ago

Logs appear.



The screenshot shows the 'Logs' page for the pod 'kubernetes-dashboard-6422902b-75665a1e-8658c5787c-xgrtd'. The logs are displayed in a terminal view, showing various system messages and requests. A 'Download logs' button is visible in the top right corner.

```

2023/03/28 07:40:24 Getting list of all jobs in the cluster
2023/03/28 07:40:24 [2023-03-28T07:40:24Z] Outcoming response to 12
2023/03/28 07:40:24 [2023-03-28T07:40:24Z] Incoming HTTP/1.1 GET /api
(MISSING)itemsPerPage=10&page=1&sortBy=d,creationTimestamp request :
2023/03/28 07:40:24 Getting list of all deployments in the cluster
2023/03/28 07:40:24 [2023-03-28T07:40:24Z] Incoming HTTP/1.1 GET /api from 127.0.0.1:
2023/03/28 07:40:24 Getting list of all cron jobs in the cluster
2023/03/28 07:40:24 [2023-03-28T07:40:24Z] Incoming HTTP/1.1 GET /api 127.0.0.1:
2023/03/28 07:40:24 Getting list of all pods in the cluster
W0328 07:40:24.827364 1 warnings.go:70] batch/v1beta1 CronJob
  
```

8. Click Download logs .

Forwarding logs to your observability system

You can forward logs from environments activated in Cloudera Data Warehouse to observability and monitoring systems such as Datadog, New Relic, or Splunk. You learn how to configure a Cloudera Data Warehouse environment for these systems.

About this task

After configuring log forwarding as described in this task, logs flow from Cloudera Data Warehouse to your system automatically. You enjoy the convenience of sorting, searching, and viewing logs on your own system instead of grepping logs from diagnostic bundles on S3 or ABFS. In addition to configuring the log forwarding, you configure removal of debug logs and text strings from the logs. You can configure log forwarding to one of the following observability systems:

- Datadog — <https://github.com/DataDog/fluent-plugin-datadog>
- Honeycomb.io — <https://docs.honeycomb.io/getting-data-in/logs/log-collectors/fluentd/>
- New Relic — <https://github.com/newrelic/newrelic-fluentd-output>
- Splunk — <https://github.com/splunk/fluent-plugin-splunk-hec> (covers both Splunk-HEC and Splunk-SCS)

You create the log forwarding configuration in valid fluentd format. The configuration is inserted into a larger fluentd configuration. All fluentd events are copied and relabeled with the new label `@cloudera_cdw`. Your custom configuration is then inserted between `<label>` tags:

```
<label @cloudera_cdw>
```

customer config goes here


```
</label>
```

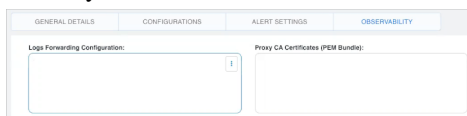
You can use any of the built-in fluentd filter, formatter, parser, or output plugins to build the custom config.

Before you begin

- Before configuring log forwarding you must [activate an AWS environment](#) or [activate an Azure environment](#) in Cloudera Data Warehouse.
- You must be [familiar with fluentd](#) and accept the responsibility of configuring log forwarding to your observability systems.

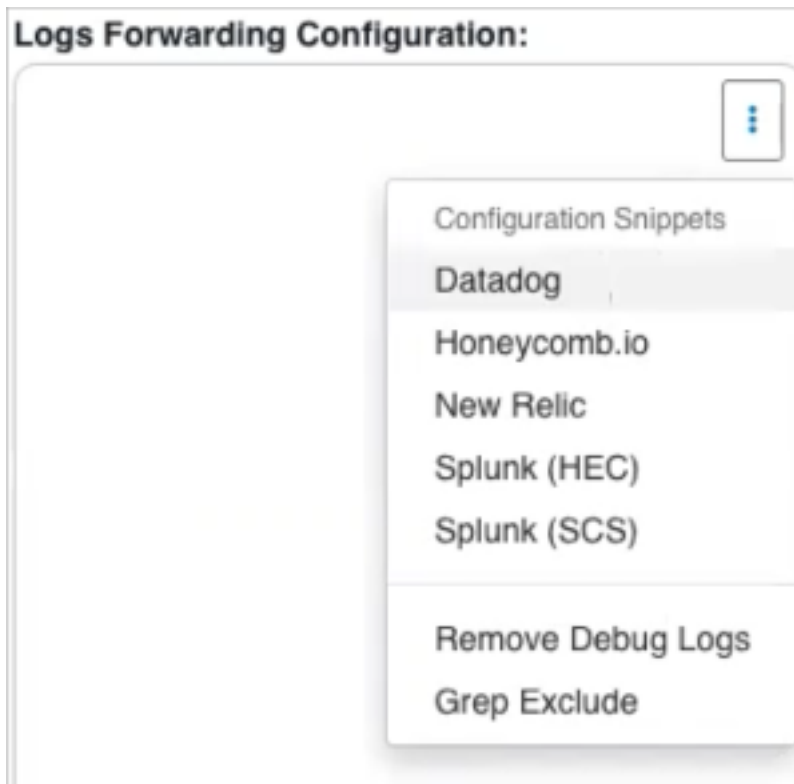
Procedure

1. In the Cloudera Data Warehouse service, go to the Environments tab.
2. Locate your environment, and click  Edit Observability .

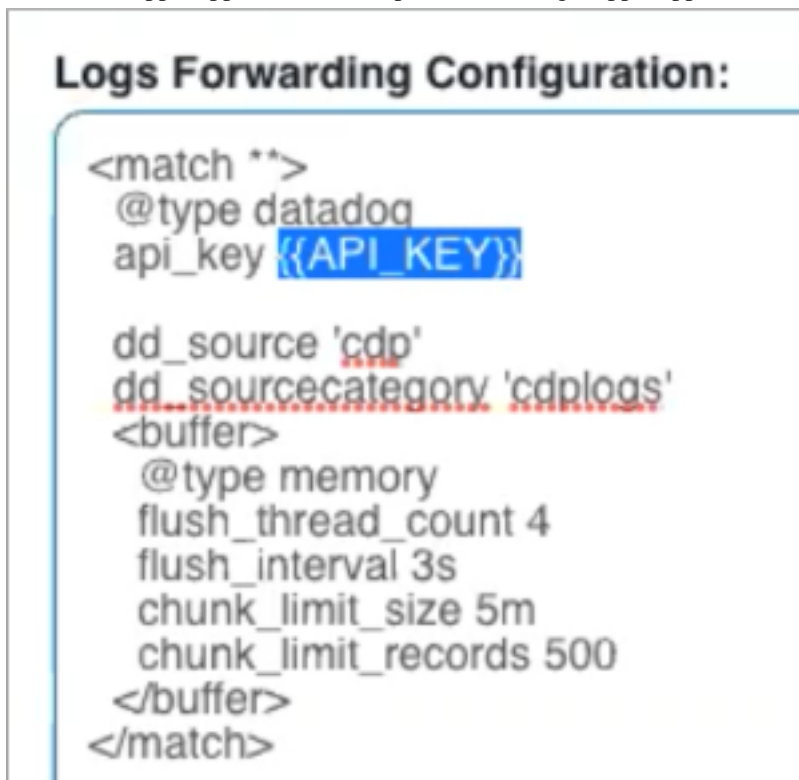


3. Decide how you want to create the fluentd config.
 - Write your own fluentd config from the ground up.
 - Use a Cloudera-provided snippet as a template to write your fluentd config.

4. In Log Forwarding Configuration, click .



5. Select one of the systems, such as Datadog, to configure.
A fluentd snippet appears. For example, the Datadog snippet appears:




6. Replace the snippet with the fluentd config you wrote from the ground up, or customize the provided snippet. For example, to customize the provided snippet replace the placeholder {{API Key}} with the actual key.

Logs Forwarding Configuration:

```
<match **>
  @type datadog
  api_key 5674465

  dd_source 'cdp'
  dd_sourcecategory 'cdplogs'
  <buffer>
    @type memory
    flush_thread_count 4
    flush_interval 3s
    chunk_limit_size 5m
    chunk_limit_records 500
  </buffer>
</match>
```

7. (Optional) If debug level log messages are not desired, add a fluentd filter to remove them: In the environment, click , and select Remove Debug Logs.

The fluentd snippet appears for removing debug logs. For example:


Logs Forwarding Configuration:

```
<filter *>
  @type grep
  <exclude>
    key log
    pattern /debug/
  </exclude>
</filter>
|
<match *>
  @type datadog
  api_key 5674465

  dd_source 'cdp'
  dd_sourcecategory 'cdplogs'
  <buffer>
    @type memory
    flush_thread_count 4
    flush_interval 3s
    chunk_limit_size 5m
    chunk_limit_records 500
  </buffer>
</match>
```

No user customization is necessary to remove debug logs.

8. (Optional) If certain log messages do not provide value for you, remove them with a fluentd grep exclude filter:

In the environment, click , select Grep Exclude, and replace {{PATTERN}} with the grep expression that matches the phrase you want to exclude.

Logs Forwarding Configuration:

```
<filter *>
  @type grep
  <exclude>
    key log
    pattern /debug/
  </exclude>
</filter>
<filter *>
  @type grep
  <exclude>
    key log
    pattern /Idontshowup/
  </exclude>
</filter>

<match *>
  @type datadog
  api_key 5674465

  dd_source 'cdp'
  dd_sourcecategory 'cdplogs'
  <buffer>
    @type memory
    flush_thread_count 4
    flush_interval 3s
    chunk_limit_size 5m
```

For more information about using Grep Exclude, see <https://docs.fluentd.org/filter/grep>.

9. If you use a proxy server for outbound traffic, provide the proxy server's CA certificates PEM bundle as described in the next task.

10. Click Apply Changes.

Cloudera Data Warehouse tests the log forwarding configuration and proxy CA certificates bundle, and saves the configuration if both are valid. An invalid log forwarding config error message appears in the event of a configuration problem. For example:

RuntimeErr with ErrCode=1042 (cause: invalid log forwarding config) [request-id: edws-internal-edcc8825]

If your configuration is valid, Cloudera Data Warehouse initiates a restart of fluentd to apply the updated config. You see the following indicators of success:

- The environment Running indicator changes, blinks Updating, and then once again says Running.
- You see logs appearing in your observability system.

Many factors affect how long it takes for forwarding to begin, but generally, the bigger your Cloudera Data Warehouse environment, the longer it takes.

Providing proxy CA certificates

If you use a TLS-terminating proxy server to inspect outbound internet traffic, you need to provide the proxy server's CA certificates bundle in PEM bundle format when you configure log forwarding.

About this task


You learn how to use the Observability tab in Cloudera Data Warehouse Environment Details to configure the Proxy CA Certificates (PEM Bundle) field.

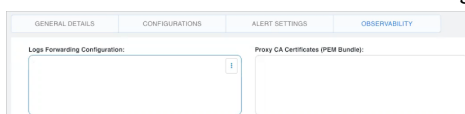
Before you begin

Before you apply the proxy CA certificate to a configuration of log forwarding, you must provide a configuration in the Logs Forwarding Configuration section of the Observability tab.

Procedure

1. In the Cloudera Data Warehouse service, go to the Environments tab.

2. Locate your environment and click  Edit Observability .
Environment details include a UI for configuring log forwarding.



3. Obtain and copy your proxy server's CA certificates PEM bundle.

4. In Proxy CA Certificates (PEM Bundle), paste the copy of the PEM bundle.

5. Click Apply Changes.

If the certificate and log forwarding configuration are valid, log forwarding begins. If the certificates are invalid, an error message occurs.

invalid proxy CA certificates bundle [request-id: edws-internal-ad92c0f3]

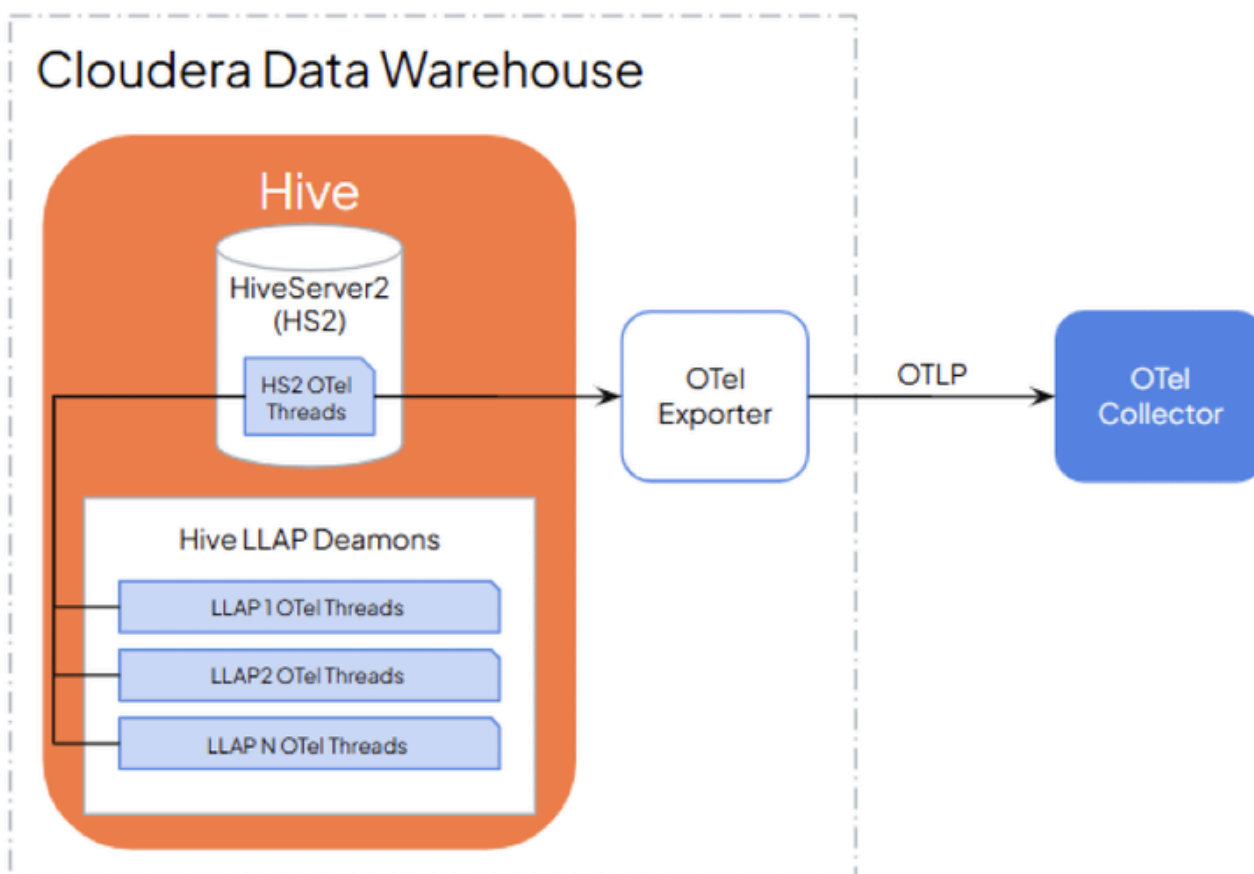
The log forwarding configuration and certificates are not saved.

OpenTelemetry support for Hive

Overview

OpenTelemetry (OTel) provides an open-source solution for collecting, processing, and exporting telemetry data, including metrics from applications. OTel helps users gain visibility into query performance and troubleshoot query failures.

As part of this offering, Hive in Cloudera Data Warehouse includes an OTel exporter that helps to collect, filter, and publish telemetry information, such as infrastructure and workload metrics, live and historical query data.



HiveServer2 and LLAP each transmit telemetry data to an OTel agent. The OTel agent independently transmits the data to a customer configured OTel collector instance for processing. The processed data can be exported and visualized through backend systems, such as Jaeger, Zipkin, Prometheus.



Note: The OTel collector is not part of the Cloudera Data Warehouse deployment. You must have your own instance of the OTel collector that is configured with backend instances (Jaeger, Zipkin, Prometheus).

HiveServer2 and LLAP integration with OTel

OTel threads function independently in both HiveServer2 and LLAP daemons. When query execution begins, these threads capture essential telemetry data, which is transmitted to an OTel collector based on a configurable recurring schedule. The collected data is then processed and forwarded to backend systems for visualization.

HiveServer2 integrates with the OTel agent to expose both query-related data and JVM metrics. A dedicated thread or service runs within HiveServer2 to handle this integration. This thread collects query details and metrics, which are then transmitted through the OTel agent. These transmitted metrics can be collected by OTel collectors for analysis and visualization.

Metrics specific to each LLAP daemon, such as JVM and memory-related statistics, are also transmitted for detailed observability.

Availability

OTel support for Hive is made effective as of the Cloudera Data Warehouse on cloud1.10.1 version. After upgrading the Cloudera Data Warehouse version, you must also upgrade existing Hive Virtual Warehouses to 2025.0.19.0 or higher versions to enable and configure the OTel integration

Related Information

[OpenTelemetry](#)

Benefits of OTel Hive Integration

This topic explains the benefits of integrating OpenTelemetry (OTel) with HiveServer2 and LLAP, focusing on enhanced observability, telemetry data insights, and optimized performance.

The integration of OpenTelemetry (OTel) with HiveServer2 and LLAP provides advanced telemetry capabilities, enabling better observability and diagnostics while maintaining optimal system performance.

Telemetry Data Exposed

OTel integration allows HiveServer2 and LLAP to expose the following telemetry data through an OTel collector:

- **Metrics:** Infrastructure and workload metrics, such as JVM memory usage, thread counts, and Operating System-related insights.
- **Live Query Data:** Tracking of active query lifecycle events, including execution times, stages, and error messages.
- **Historical Query Data:** Detailed query execution metadata for analysis and diagnostics.

Performance Benefits

The OTel integration is designed to enhance observability while ensuring HiveServer2 and LLAP maintain optimal performance.

Optimized for Minimal Impact: Integrating OTel with HiveServer2 and LLAP ensures seamless performance. By utilizing an independent thread or service to collect, translate, and expose metrics already tracked by HiveServer2, the integration ensures query execution remains smooth, without any noticeable delays or disruptions.

Scalable for Future Metrics: The system is well-prepared to handle new metrics if introduced. While tracking additional metrics may require some extra resources, this scalable design ensures the system remains efficient, with any potential impact being manageable and dependent on your specific use case.

Efficient Memory Usage: The OTel thread is optimized for minimal memory consumption, efficiently managing the receive, translate, and expose phases. The memory usage is small and well within acceptable limits, ensuring it doesn't affect overall system performance, even during extended operations.

Configuring OTel in HiveServer2

Learn how you can enable OTel in HiveServer2 and configure certain properties that will enable you to optimize the data collection.


Before you begin

- Ensure that you are on Cloudera Data Warehouse 1.10.1 or higher version.
- Ensure that the Hive Warehouse version is on 2025.0.19.0 or higher version.

Procedure

1. Log in to the Cloudera web interface and navigate to the Cloudera Data Warehouse service.
The Overview page is displayed.

- From the Overview page, click the Virtual Warehouses tab, identify the Hive Virtual Warehouse that you want to

configure, and then click  to edit.

- In the Virtual Warehouse details page, click **Configurations Hiveserver2** and then select **hive-site** from the Configuration files drop-down.

- Search for **hive.otel** and modify the values as required:

- hive.otel.metrics.frequency.seconds** (Default: 0s)

Specifies the frequency at which telemetry data is transmitted to the OTel collector. By default, the value is set to 0 seconds indicating that OpenTelemetry data collection is disabled. Enter a value greater than 0 to enable OpenTelemetry.

If the value is 5s. This indicates that telemetry data is transmitted to the OTel collector every 5 seconds.

- hive.otel.collector.endpoint**

Specifies the endpoint where all the OpenTelemetry Protocol (OTLP) traces and metrics are transmitted. The endpoint represents the address of an OTel collector. The endpoint must be a valid URL with https scheme.

`https://<otel-collector-host>:<port>`



Important: Cloudera recommends using https to ensure secure data transmission.

- hive.otel.exporter.timeout** (Default: 10m)

Specifies the maximum time allowed for the OTel agent to complete a transmit operation. The transmit operation times out if it exceeds the specified time.

- hive.otel.retry.initial.backoff** (Default: 10s)

Specifies the initial time delay before attempting to retry a failed transmit operation. The value serves as the starting point for the exponential backoff strategy.

- hive.otel.retry.max.backoff** (Default: 1m)

Specifies the maximum time that the OTel agent should wait between retries. This sets an upper limit on the backoff interval ensuring that retry export operations do not exceed the specified duration even with exponential backoff.

- hive.otel.retry.backoff.multiplier** (Default: 5f)

Specifies the factor by which the retry interval increases after every failed attempt. This determines how much the backoff interval increases after each failed attempt, following an exponential backoff strategy.

- Click **Apply Changes** and restart the Hive Virtual Warehouse.

Telemetry data exposed to OTel collector

HiveServer2 transmits telemetry data related to live queries, completed queries, task-level details, JVM metrics related to memory usage and thread count, and Operating System level statistics.

Query and Task related insights

Live queries

The following metrics related to live HiveServer2 queries are transmitted to the OTel collector:

Metrics	Description
QueryId	Represents the unique identifier for the query.
QueryString	The SQL statement of the query.
UserName	The user who submitted the query.
ExecutionEngine	The query execution engine used to process the query, typically Tez.

Metrics	Description
ErrorMessage	Errors encountered during query execution.

Completed queries

The following metrics related to completed HiveServer2 queries are transmitted to the OTel collector:

Metrics	Description
QueryId	Represents the unique identifier for the query.
QueryStartTime	Timestamp when the query execution started.
EndTime	Timestamp when the query execution completed.
OperationId	Unique identifier for tasks related to query execution. For example, a0fe8acc-6b9a-4f54-8537-f7b3bf7dea72
OperationLogLocation	Path to the local log file for additional query log entries.
ErrorMessage	Errors encountered during query execution.
ExplainPlan	Output showing how the engine executed the query.
FullLogLocation	Location of the complete application logs related to the query.
Running	Indicates whether the query is currently in progress.
Runtime	Duration of the query execution in seconds or minutes.
UserName	The user who submitted the query.
ExecutionEngine	The query execution engine used to process the query, typically Tez.
State	Current state of the query (e.g., RUNNING, SUCCESS, ABORTED, or FAILED).
SessionId	Identifier for the session in which the query was executed.

Task-level details

The following metrics related to tasks are transmitted to the OTel collector:

Metrics	Description
TaskId	Unique identifier for each task.
Name	Task types such as MAPRED (Mapper/Reducer tasks), DEPENDENCY_COLLECTION, or STATS TASK. If multiple tasks exist, integers are appended.
TaskType	Representation with unique values while execution such as MAPRED, DEPENDENCY_COLLECTION, MOVE, or STATS.
Status	Current task status (e.g. Success).
StatusMessage	Status with detailed task information
ExternalHandle	DAGID, used for query optimization and execution.
ErrorMsg	Error details if the task failed.
ReturnValue	Task result. A value of 0 indicates success, while negative integers indicate an issue.
BeginTime	Time when the task execution started.
ElapsedTime	Time spent executing the task.
EndTime	Time when the task execution finished.

Java Virtual Machine (JVM) metrics

HiveServer2 collects various JVM metrics, including:

- Memory usage: Data such as heap and non-heap memory usage.
- Thread count: Counts of threads in different states (for example, runnable, waiting).
- OS-Level statistics: CPU load, memory size, and swap space details.

Metrics related to memory usage

The following metrics related to JVM heap and non-heap memory usage are transmitted to the OTEL collector:

Metrics	Description
memNonHeapUsedMGauge	Size (in MB) of non-heap memory currently used by the JVM.
memNonHeapCommittedM	Amount of non-heap memory (in MB) reserved by the JVM for internal use.
memNonHeapMaxM	Maximum allowable size (in MB) for non-heap memory.
memHeapUsedM	Size (in MB) of heap memory currently used by the JVM.
memHeapCommittedM	Size (in MB) of heap memory reserved by the JVM.
memHeapMaxM	Maximum allowable size (in MB) for heap memory.
memHeapMaxM	Maximum memory available (in MB) to the JVM.

Metrics related to thread count

The following metrics related to the JVM threads in different states (runnable, waiting) are transmitted to the OTEL collector:

Metrics	Description
threadsNew	Number of threads currently in the "NEW" state within a Java application.
threadsRunnable	Number of threads ready to run or currently executing on the CPU.
threadsBlocked	Number of threads waiting to acquire a lock.
threadsWaiting	Number of threads waiting indefinitely for a signal from another thread.
threadsTimedWaiting	Number of threads waiting for a specific duration before proceeding.
threadsTerminated	Number of threads where execution is completed.

OS-level statistics

The following OS related metrics, such as CPU load, memory size, swap space details are transmitted to the OTEL collector:



Note: These metrics are transmitted only for Unix-based operating systems.

Metrics	Description
systemCpuLoad	Measures the CPU load of the host machine, container, or pod.
committedVirtualMemorySize	Amount of allocated memory, including both physical RAM and virtual memory reserved by the operating system.

Metrics	Description
processCpuTime	Total CPU time consumed by a specific process or thread.
freePhysicalMemorySize	Amount of unused physical memory (RAM) available to processes and the operating system, container, or pod.
freeSwapSpaceSize	Amount of swap space currently available.
totalPhysicalMemorySize	Total installed physical memory (RAM) in the system.
processCpuLoad	Percentage of CPU load consumed by a specific process.

Example Visualizations through OTEL backend

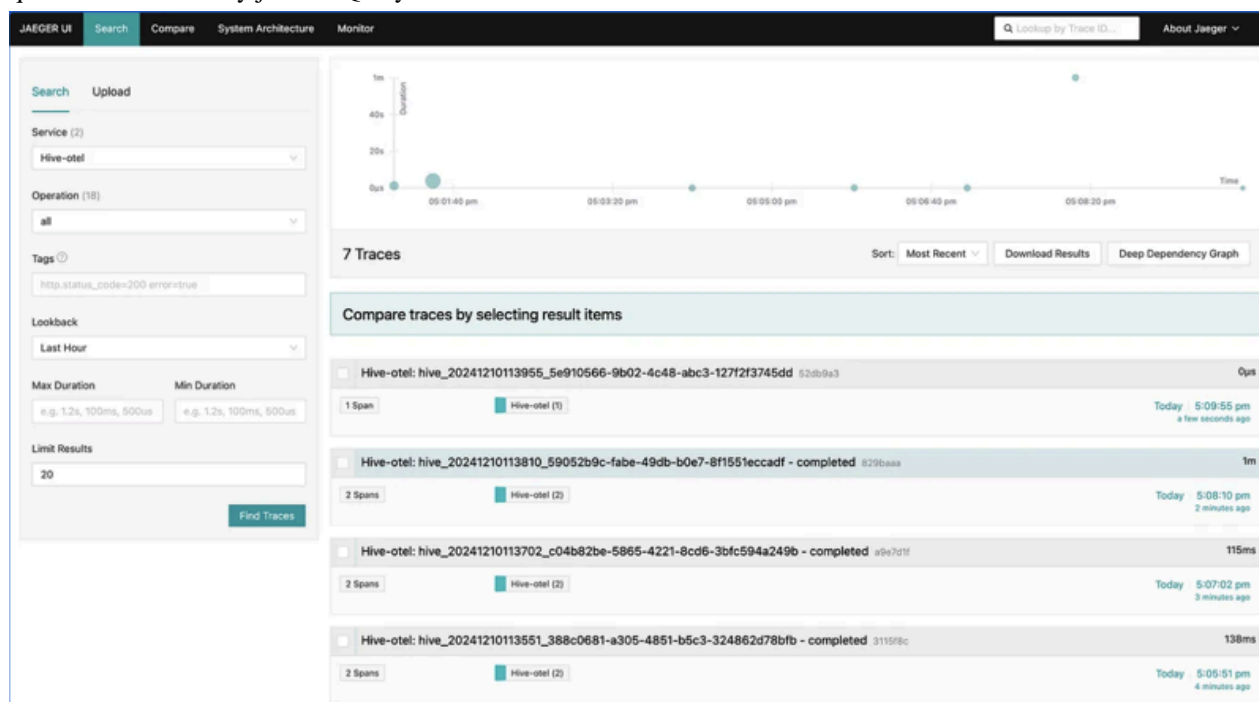
This section provides some examples of OTEL backend systems that allow you to view the telemetry data that is processed and exported from an OTEL collector.



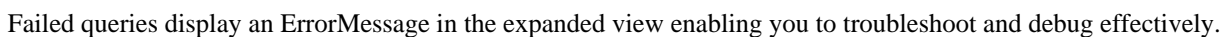
Important: The example backend systems shown here are just for representation. Along with having your own instance of the OTEL collector, you must also have your own instances of Jaeger, Zipkin, Prometheus and Grafana to view the telemetry data.

Visualizing through Jaeger

The following image represents a Jaeger UI displaying traces for 4 queries out of which 1 is a live query and the remaining 3 are completed queries. The completed queries are denoted by the completed suffix and the running queries are denoted by just the Query ID and do not have a suffix.



You can click on a trace to view more details about the attributes and tasks. This helps you understand task breakdowns and performance aspects of individual query components.



Limitation of OpenTelemetry support for Hive

Learn about the current limitations of OTel integration, including its focus on metrics and events, while future-proofing for logs. Understand the fixed nature of event data and its implications for query observability.

Telemetry Data Scope

The scope of telemetry data is currently limited to only metrics and events. Logs and traces are under consideration for a future release.

Fixed Event Data

The events being sent through an OTel agent are not configurable and can include Personally Identifiable Information (PII) within the SQL statement of the query.